LOAN DOCUMENT

PHOTOGRAPH THIS SHEET

DTIC ACCESSION NUMBER

LEVEL

INVENTORY

DOCUMENT IDENTIFICATION

DISTRIBUTION STATEMENT

HANDLE WITH CARE

ACCESSION FOR
NTIS GRA&I
DTIC TRAC
UNANNOUNCED
JUSTIFICATION

BY

DISTRIBUTION/

AVAILABILITY CODES

DISTRIBUTION	AVAILABILITY AND/OR SPECIAL
A-1	

DISTRIBUTION STAMP

DATE ACCESSIONED

DATE RETURNED

19981223 011

DATE RECEIVED IN DTIC

REGISTERED OR CERTIFIED NUMBER

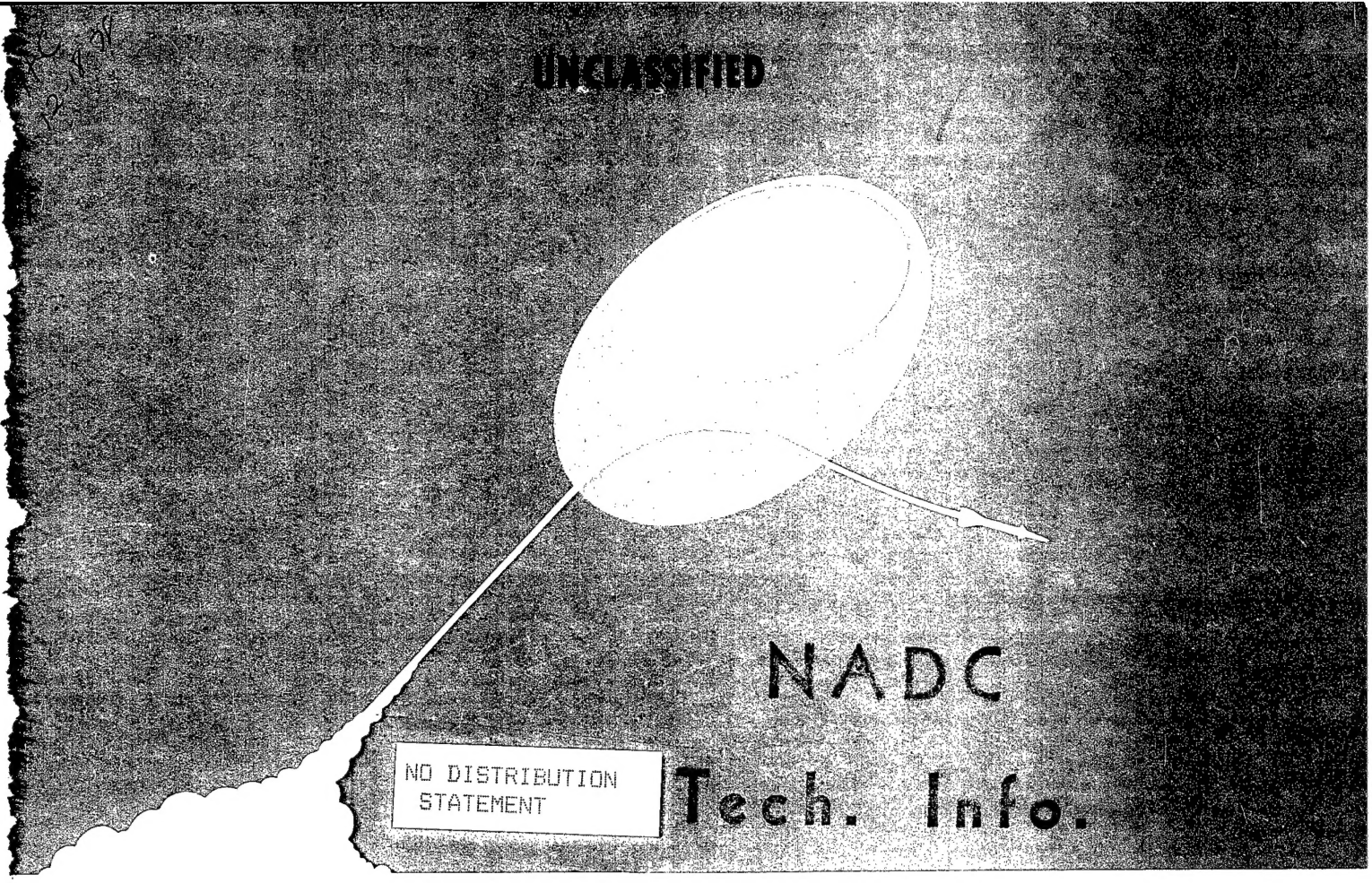
PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-FDAC

DTIC FORM 70A JUN 90

DOCUMENT PROCESSING SHEET

PREVIOUS EDITIONS MAY BE USED UNTIL STOCK IS EXHAUSTED.

LOAN DOCUMENT



APPENDIX 10

SORTER MESSAGE PROCESSING

FINAL SOFTWARE REPORT

DATA ITEM NO. A005

INTEGRATED ELECTRONIC WARFARE SYSTEM ADVANCED DEVELOPMENT MODEL (ADM)

7800787-11

PREPARED FOR:

NAVAL AIR DEVELOPMENT CENTER
WARMINSTER, PENNSYLVANIA

CONTRACT N62269-75-C-0070

RAYTHEON
ELECTROMAGNETIC SYSTEMS DIVISION

1 OCTOBER 1977

APPENDIX 10

SORTER MESSAGE PROCESSING DESIGN SPECIFICATION

FINAL SOFTWARE REPORT

DATA ITEM A005

INTEGRATED ELECTRONIC WARFARE SYSTEM (IEWS)
ADVANCED DEVELOPMENT MODEL (ADM)

Contract No. N62269-75-C-0070

Prepared for:

Naval Air Development Center
Warminister, Pennsylvania

Prepared by:

RAYTHEON COMPANY
Electromagnetic Systems Division
6380 Hollister Avenue
Goleta, California 93017

1 OCTOBER 1977

RAYTHEON

RAYTHEON COMPANY
LEXINGTON, MASS. 02173

CODE IDENT NO.	SPEC NO.
49956	53959-GT-0755
	SHEET 1 OF 72 REV

TYPE OF SPEC

COMPUTER SUBPROGRAM DESIGN DOCUMENT

TITLE OF SPEC

SORTER MESSAGE PROCESSING, IEWS, ADM

FUNCTION	APPROVED	DATE	FUNCTION	APPROVED	DATE
WRITER	Thomas L. Chernosky	13 Oct 76			

REVISIONS

CHK	DESCRIPTION	REV	CHK	DESCRIPTION	REV

REVISION																										
SHEET NO.																										
REV STATUS OF SHEETS	REVISION																									
	SHEET NO.																									

10-1348 (11-68) VELLUM PRINTED IN U.S.A.
10-1348 (11-68) FILM

TABLE OF CONTENTS

1.0 SCOPE

1.1 IDENTIFICATION

This document specifies the implementation of the Signal Sorter Message Processing Functional Group (SOFG) of the SC Operational Software resident in the Classification Processor.

1.2 SUBPROGRAM TASKS

1.2.1 Signal Sorter Message Processing Driver

SODR shall decode messages received by the SC from the Signal Sorter (SS). The appropriate message processor shall then be called to do the actual message processing.

1.2.2 Throttle Alert Processing

SOTHR shall process SS Message Op-Code X'84', the Throttle Alert message.

1.2.3 Confirm File Creation Processing

SOCFC shall process SS Message Op-Code X'85', the Confirm File Creation message.

1.2.4 System Management 1 Processing

SOSM1 shall process the following SS Message Op-Codes:

X'89' IB less than 1/4 full
X'8A' IB greater than 3/4 full
X'8B' Files full
X'8C' Throttle files full

1.2.5 Long Pulse Messages

SOLP shall process SS Message Op-Code X'88', the Long Pulse Parameter message.

1.2.6 ALR-5Ø Messages

SOALR shall process SS Message Op-Code X'8F', the ALR-5Ø message.

1.2.7 Update Processing

SOUP shall process SS Message Op-Code X'8Ø', the PTDW message.

1.2.8 MFF Processing

SOMFF shall process SS Message Op-Code X'92', the Multifrequency Flags message.

1.2.9 Deletion Processing

SODEL shall process SS Message Op-Code X'87', the Inactive File Alert.

1.2.10 New Emitter Processing

SONE1 shall process SS Message Op-Code X'8'', the New Emitter Alert message.

1.2.11 Sorter Instrumentation

SOINS shall process the following SS Message Op-Codes:

X'82' Cam File Dump
X'83' AOA Readout
X'86' Error Alert
X'8D' Bus Hung
X'8E' Watchdog Timer
X'9Ø' NPDW Message
X'91' Memory Dump
X'93' BIT Status

2.0 APPLICABLE DOCUMENTS

The following documents, of the exact issue shown, form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of the Computer Program Design Specification for the Integrated Electronic Warfare System (IEWS) Advanced Development Model (ADM) Program shall be considered superseding requirements.

2.1 COMPUTER PROGRAM PERFORMANCE SPECIFICATION

Computer Program Performance Specification for the Integrated Electronic Warfare System (IEWS) Advanced Development Model (ADM) Program (U), Raytheon Company, Electromagnetic Systems Division, (Number 061290529), (date 1 June 1976), (classification U).

2.1.1 Applicable CPPS Paragraphs

Sorter Message Driver	Not specified explicitly
Throttle Alert *	3.3.2.1.2.5
File Creation	Not specified explicitly
System Management 1	" " "
Long Pulse	3.3.2.1.2.1.3
ALR-50	3.3.2.3
Update Processing	3.3.2.1.2.2
MFF Processing	3.3.2.1.2.4
Deletion Processing	3.3.2.1.2.3
NE Processing 1	3.3.2.1.2.1
Sorter Instrumentation	3.2.4.3

2.2 COMPUTER PROGRAM DESIGN SPECIFICATION

Computer Program Design Specification for the Integrated Electronic Warfare System (IEWS) Advanced Development Model (ADM) Program (U), Raytheon Company, Electromagnetic Systems Division, (Number 53959-GT-0750), (date TBD), (classification U).

2.3 DATA BASE DESIGN DOCUMENT

The Common Data Base Design Document, System Controller Unit, IEWS, ADM, document No. 53959-GT-0751, shall apply to this subprogram.

2.4 MISCELLANEOUS DOCUMENTS

The following documents shall apply to this subprogram:

Document No.	Document Title
53959-GT-0756	Computer Subprogram Design Document, Executive, IEWS
53959-JK-1002	Interface Control Document, System Controller-Sorter
TBD	Computer Subprogram Design Document, Emitter Classification 1, IEWS
53959-GT-0759	Computer Subprogram Design Document, Data Extraction, IEWS
WS-8506 Revision 1, 1 November 1971	Requirements for Digital Computer Program Documentation

3.0 REQUIREMENTS

3.1 SUBPROGRAM DETAILED DESCRIPTION

3.1.1 Signal Sorter Message Processing (SODR)

SODR shall be the driver routine of the SS Message Processing Functional Group. The EXEC shall pass to SODR a pointer to the SS Message Flag word in the X-register. SODR shall increment the pointer and then use it to retrieve the op-code from the left byte of word Ø of the message. X'8Ø' shall then be subtracted from the op-code, in order to compute an index, i. This index, which exists only in the A-register, shall then be verified to be in the range $Ø \leq i \leq X'13'$. If i is out of range, control shall be transferred to label SOD8Ø where an instrumentation message shall be sent to the EXEC and a subroutine return performed to return control to the EXEC. If i is in the specified range, the value of i (A-register) shall be added to the address of the SS Message Processing Table (SOMPT) and the result stored in the B-register. The effective address (B-reg) shall be used indirectly to call one of the SS Message processing routines, whose list of symbolic names constitute SOMPT. All of the message processing routines shall be passed the same data:

1) The address of the SS message in the X-register
2) The Sorter file number in the A-register

After the individual message processing routine has completed its task, control shall be returned to the Sorter Message Driver. If the message processing routine (e.g., New Emitter Processing 1) has returned to the driver via the first return, the Executive message function shall be called to output the analysis request message (label SOD9Ø) and then control shall be returned to the EXEC. The message processing routine shall pass to SODR a pointer to the message buffer in the X-register and this shall be passed

to the EXEC in the X-register. If the message processing routine has returned via the second return, control shall be immediately returned to the EXEC via a subroutine return.

3.1.2 Throttle Alert Processing (SOTHR)

SOTHR shall be called by SODR to process Sorter Throttle Alert messages (Op-Code = X'84'). Subroutine SOGET shall be immediately called to convert the Sorter File Number (SFN) into the address of a Emitter Track File entry (EF entry). The Throttle bit (EFTH) shall be set in the EF entry. The Throttle File Number (TFN) shall be retrieved from the Sorter message data and stored in the EF entry as EFTFN. The Reduction Factor (RF) shall also be retrieved from the message and stored in the EF entry. Control shall be transferred to label SONAX (see 3.1.3.2) to return to the driver (Return 2 - No Analysis Request).

3.1.2.1 Convert Emitter File Number to EF Entry Address (SOGET) -

Subroutine SOGET shall be passed the emitter file number in the right byte of the A-register. The left byte shall be zeroed out and the result multipled by 16 (the length of an EF entry). The result shall be added to the starting address of the EF table. The result, which is the EF entry address, shall be returned in the B-register to the calling routine.

3.1.3 Confirm File Creation Processing (SOCFC)

SOCFC shall be called by SODR to process Sorter Confirm File Creation messages (Op-code = X'85'). SOMSG, a sorter instrumentation message buffer (see Figure 5) shall be used to store the sorter message data. At SOC1Ø the message number (SOMNO) shall be set to 5. SODR shall pass to SOCFC a pointer to the Sorter Message Op-Code word in the X-register. This pointer shall be decremented and used to retrieve

the sorter message word count from the flag word. The word count in SOMSG shall be set to the value of sorter word count +1 and a pointer to the next available word in SOMSG (word 3) shall be loaded into the B-register and processing continues at SOC5Ø.

3.1.3.1 SOC5Ø

The data from the Sorter message shall be copied into the SOMSG buffer. The EXEC shall then be called to output the message. Processing continues at SONAX.

3.1.3.2 SONAX

A jump to SONAX shall be executed whenever a Sorter message processing routine executes the second return back to the driver (no analysis message). The return address on the stack shall be incremented and control returned to the driver.

3.1.4 System Management 1 (SOSM1)

SOSM1 shall be called to process the following Sorter messages:

X'89' IB less than 1/4 full
X'8A' IB greater than 3/4 full
X'8B' Files full
X'8C' Throttle files full

The message number, SOMNO, shall be set to 6 in SOMSG, a System Management 1 message buffer (see Figure 6). The data from the Sorter message shall then be copied into the message buffer. The Executive message function shall then be called to output the message and control shall be returned to the driver via a no-analysis return.

3.1.5 Long Pulse Processing (SOLP)

SOLP shall be called by SODR to process Sorter Long Pulse Parameter messages (Op-Code = X'88'). The name SOLP is simply a synonym for SOCFC. See 3.1.3.

3.1.6 ALR-50 Processing (SOALR)

SOALR shall be called by SODR to process Sorter ALR-5Ø messages (Op-Code = X'8F'). The name SOALR is simply a synonym for SOCFC. See 3.1.3.

3.1.7 Update Processing (SOUP)

SOUP shall be called by SODR to process Sorter Pulse Train Descriptor Word (PTDW)messages (Op-Code = X'8Ø'). This routine shall be passed a pointer to the Sorter Message Op-Code word in the X-register and the Sorter File Number (SFN) in the A-register. SOGET shall be immediately called to transform the SFN into an Emitter Track File entry (EF entry) address. SOGET shall return the address in the B-register. The identification code (EFID) shall be retrieved from the EF entry. If EFID is the unclassified code, control shall be transferred to label SONAX (see 3.1.3.2) to return control to the driver (no analysis message return). If EFID is the NOFA1 code the NOFA processing routine (SONA2) shall be called and then control shall be transferred to SONAX for both SONA1 returns (Emitter not within limits and emitter within limits). If EFID is the NOFA2 code, the NOFA2 processing 1 routine (SON21) shall be called. If SON21 performs an analysis request return, the X-register shall contain a pointer to the analysis request message and control shall be transferred to SOU99. Otherwise, control shall sent to SONAX. Finally, if EFID is anything else, the EOC Processing 1 routine (SOOC1) shall be called. If SOOC1 performs an analysis request return the X-register shall contain a pointer to the analysis request message and control shall be transferred to SOU99. Otherwise, control shall be sent to SONAX.

3.1.7.1 SOU99

A subroutine return shall be performed to send control back to the driver. Note that this exit shall result in an analysis request return to SODR being performed.

3.1.7.2 NOFA1 Processing (SONA1)

SONA1 shall be called to process Sorter PTDW's on NOFA1 classified emitters. SONA1 shall be passed as input:

3.1.7.2 -continued-

1) A pointer to the SS message op-code word in the X-register.

2) A pointer to the EF entry in the B-register.

The Limit Test (SOLT) shall be called to see if any of the critical track parameters have changed significantly (Frequency, PRI, and Pulse Width). The input to SOLT shall be the same as the input to SONA1. If SOLT performs a Not-within-limits return (return no. 1), processing shall continue at SONA5. Otherwise, the return-to-SOUP address on the stack shall be incremented (in anticipation of performing an "Emitter is within limits" return - return no. 2). Processing shall then continue at SONA8.

3.1.7.2.1 SONA5 - SOLB shall be called to transfer the Sorter message frequency (TFREQ), pulse width (TPW) and average PRI (TPRIA/2 + TRRIB/2) into the EF entry. The input to SOLB is the same as the input to SONA1. After SOLB returns, the Emitter File Number shall be retrieved from the Sorter message and stored as SOEFN in SOCLM, a message buffer which has the format of a classification message (to CP) (See Figure 2). The EXEC shall then be called to output the classification message. Processing continues at label SONA8.

3.1.7.2.2 SONA8 - SOLC (Update Azimuth Links) shall be called to update the azimuth links for the emitter, if necessary and then SOLA shall be called to transfer the Sorter message azimuth (TAZ) and peak amplitude (TPAMP) into the EF entry. The input to SOLC and SOLA shall be the same as for SONA1. Control shall then be returned to the calling routine. Note that either one of two returns shall be possible:

1) Emitter is within limits, or
2) Emitter is not within limits.

3.1.7.3 NOFA2 Process 1 (SON21)

SON21 shall be called by SOUP if the identification code of the emitter being updated is NOFA2. SON21 shall be paased as input:

1) A pointer to the Sorter message op-code word in the X-register.
2) A pointer to the EF entry in the B-register.

NOFA1 processing (SONA1) shall be immediately called. If SONA1 returns via the first return (not within linits), processing continues at label SON2X. Otherwise, the peak amplitude value (TPAMP) shall be retrieved from the Sorter message and compared against the value of A_T. If TPAMP is the greater, processing continues at SON2X. Otherwise, the return address on the stack shall be incremented so that a no-analysis return (return no. 2) is performed to send control back to the calling routine.

3.1.7.4 SON2X

The pointer to the scan analysis request buffer SOA21 (see Figure 2) shall be loaded into the X-register and control returned to the calling routine via the analysis request return (return no. 1).

3.1.7.5 EOC Process 1 (SOOC1)

SOOC1 shall be called by SOUP if the identification code of the emitter being updated is not unclassified, NOFA1, or NOFA2. SOOC1 shall be passed as input:

1) A pointer to the Sorter message op-code word in the X-register.
2) A pointer to the EF entry in the B-register.

3.1.7.5 -continued-

SOLB shall be immediately called to transfer the Sorter message frequency (TFREQ), pulse width (TPW), and average PRI (TPRIA/2 + TPRIB/2) into the EF entry. SOLC shall then be called to update the azimuth links for the emitter, if necessary. SOLA shall then be called to transfer the Sorter message azimuth (TAZ) and peak amplitude (TPAMP) into the EF entry. The input to SOLB, SOLC, and SOLA shall be the same as the input to SOOC1.

The Emitter File Number (EFN) shall be retrieved from the Sorter message and saved in SOAC1, the EOC Process 1 Analysis Request message buffer, and in SOUPM, the update message buffer. The Level 1 Search routine (see Emitter Classification 1 CSDD) shall then be called (with the emitter file number in the X-register) in an attempt to classify this emitter. If there are no candidates (Level 1 Search return no. 1), processing shall continue at label SOO1Ø. If there are candidates (Level 1 Search return no. 2), a pointer to the candidate list in the Common Data Base shall be returned in the X-register. This pointer shall be saved in the SOAC1 Analysis Request message. Scan Test 1 (see Emitter Classification 1 CSDD) shall then be called (with the emitter file number in the X-register) to determine if analysis is required. If no analysis is required (Scan Test 1 return no. 1), the analysis wanted bit (SOAW) shall be cleared in the SOAC1 Analysis Request message buffer and processing continues at SOO8Ø. If analysis is required (Scan Test 1 return no. 2), the SOAW bit shall be set and processing continues at SOO8Ø.

3.1.7.5.1 SOO8Ø - The pointer to the SOAC1 analysis request message buffer shall be loaded into the X-register and control returned to the calling routine (return no. 1 to SOUP - analysis wanted).

3.1.7.5.2 SOO1Ø - The platform link pointer (EFPLNK) shall be retrieved from the EF entry. If there is platform linkage (i.e., EFPLNK is not equal to the EFN), the Delete Link Processing routine (DUMMY) shall be called to remove the linkage. The EXEC shall then be called to output the uptate message. The return address on the stack shall be incremented and control returned to the calling routine (return no. 2 to SOUP - no analysis wanted).

3.1.7.6 Limit Test (SOLT)

Subroutine SOLT shall be used to determine if certain critical parameters (frequency, average PRI, and pulse width) of an emitter have changed since the last update. TFREQ shall be retrieved from the Sorter message. The value the EF entry frequency (EFFREQ) shall be retrieved and subtracted from TFREQ. If the absolute value of the difference is not less than the value of Δ FREQ, processing continues at label SOLT9. Otherwise, TPRIA and TPRIB shall be retrieved from the Sorter message. The average of these two PRI's shall be calculated. The value of the EF entry average PRI (EFAVPI) shall be retrieved and subtracted from the Sorter average. If the absolute value of the difference is not less than the value of ΔPRI, processing continues at label SOLT9. Otherwise, TPW shall be retrieved from the Sorter message. The value of the EF entry pulse width (EFPW) shall be retrieved and subtracted from TPW. If the absolute value of the difference is not less than the value of PW, processing continues at label SOLT9. Otherwise, the emitter has not changed significantly (with respect to frequency, average PRI, and pulse width) and the return address on the stack shall be incremented so that return no. 2 (emitter within limits) is executed.

3.1.7.6.1 SOLT9 - A subroutine return shall be performed. Two returns are possible:

1) Emitter is not within limits.
2) Emitter is within limits

Return no. 1 shall be performed unless the return address on the stack has been incremented.

3.1.7.7 Load TAZ and TPAMP into EF Entry (SOLA)

SOLA shall be called with the following input:

1) X-register contains a pointer to the Sorter message op-code word.

2) B-register contains a pointer to the EF entry.

The Sorter message azimuth (TAZ) shall be retrieved and stored in the EF entry as EFAZ. The Sorter message peak amplitude (TPAMP) shall be retrieved and stored in the EF entry as EFPAMP. Control shall then be returned to the calling routine.

3.1.7.8 Load TFREQ, Average TPRI, and TPW into EF Entry (SOLB)

SOLB shall be called with the same input as SOLA (see 3.1.7.7). The Sorter message frequency (TFREQ) shall be retrieved and stored in the EF entry as EFFREQ. The Sorter message PRIA (TPRIA), and PRIB (TPRIB) shall be retrieved. The average of the two values shall be computed and stored in the EF entry as EFAVPI. The Sorter message pulse width (TPW) shall be retrieved and stored in the EF entry as EFPW. Control shall then be returned to the calling routine.

3.1.7.9 Update AZ Links (SOLA)

SOLC shall be called with the same input as SOLA (see 3.1.7.7). The Sorter message azimuth (TAZ) shall be retrieved and compared to the current azimuth value in the EF entry. If equal, control shall be returned to the calling routine. If not equal, SODAL shall be called (with a pointer to the EF entry in the B-register) to delete any old azimuth links for this emitter. Then SOIAL shall be called (with the EF entry pointer in the B-register and the Sorter

message azimuth (TAZ) in the A-register) to insert this emitter into an azimuth chain (or to create a new azimuth chain, if necessary). Control shall then be returned to the calling routine.

3.1.7.10 Delete Azimuth Links (SODAL)

SODAL shall be used to delete an emitter track from an azimuth chain. The input shall be a pointer to the EF entry in the B-register. The emitter file number (EFN) shall be derived from the EF entry pointer by subtracting the EF table base address and then dividing by 16. The forward azimuth link (EFLNK) and the backward azimuth link (EFBLNK) of the EF entry shall be examined in order to select one of the following cases:

Case 1: Emitter is the "top" of an azimuth chain, i.e., the last emitter added to the chain. This shall be the case if EFLNK is equal to EFN, i.e., EFLNK points to itself.

Case 2: Emitter is the "bottom" of an azimuth chain, i.e., the first emitter added to the chain. This shall be the case if EFLNK is not equal to EFN and EFBLNK is equal to EFN.

Case 3: Emitter is in the "middle" of an azimuth chain, i.e., neither first nor last. This shall be the case if EFLNK is not equal to EFN and EFBLNK is not equal to EFN.

Figure 1 shows an azimuth chain of three emitter tracks.

3.1.7.10.1 Delete Azimuth Links Case 1 ("Top") - At label SODA2, EFBLNK of the EF entry shall be retrieved and saved temporarily (on stack). Then EFBLNK shall be set to EFN to perform the deletion in the "backward" direction. Then, the azimuth link table (AZ) shall be searched sequentially for an entry which has an azimuth link (AZLNK) equal to EFN. Once found, the old value of EFBLNK shall be retrieved (from stack) and compared to EFN. If equal, the chain consisted of a single emitter and the deletion in the

"forward" direction shall consist of clearing the active bit in the azimuth link table entry (AZACT). If not equal, the azimuth chain consists of more than a single emitter and the deletion in the "forward" direction shall consist of setting the azimuth link pointer in the azimuth link table entry (AZLNK) to the old value of EFBLNK. Control is then returned to the calling routine.

3.1.7.10.2 Delete Azimuth Links Case 2 ("Bottom") - At label SODA6, EFLNK of the EF entry shall be retrieved and saved temporarily. Then EFLNK shall be set to EFN. Then the emitter file number temporarily saved shall be retrieved and converted into an EF entry address (by calling subroutine SOGET - see 3.1.2.1). EFBLNK for this EF entry shall be set to the temporarily saved EFN. Control is then returned to the calling routine.

3.1.7.10.3 Delete Azimuth Links Case 3 ("Middle") - At label SODA8, EFLNK and EFBLNK of the EF entry shall be temporarily saved on the stack (for brevity, the values shall be called TEMPI and TEMP2, respectively. EFLNK and EFBLNK shall then be set to the emitter file number to initialize the azimuth links for this EF entry. TEMP1 shall then be converted into an EF entry address by calling SOGET (with TEMP1 in the A-register). Then EFBLNK for this EF entry shall be set to TEMP2. TEMP2 shall then be converted into an EF entry address by calling SOGET (with TEMP2 in the A-register). Then EFLNK for this EF entry shall be set to TEMP1. The above operations result in the emitter track being deleted from the azimuth chain by making the azimuth-linked emitters "on either side" point to each other. Control is then returned to the calling routine.

3.1.7.11 Insert Azimuth Links (SOIAL)

SOIAL shall be called to insert an emitter track into an azimuth chain or start the chain if none exists. The input shall be a pointer to the EF entry in the B-register and the value of track azimuth (TAZ) from the Sorter

message in the A-register. The emitter file number (EFN) shall be derived from the EF entry pointer by subtracting the EF table base address and then dividing by 16. A pointer to an Azimuth Link Table entry (AZ entry) shall be calculated by adding the value of TAZ to the base address of the AZ table. The active bit (AZACT) shall be retrieved from the AZ entry. If AZACT is set, an azimuth chain already exists for this angle cell and processing continues at SOIA5. If AZACT is zero, no chain exists for this angle cell. To start the chain, the, AZACT for this AZ entry shall be set to 1 and AZLNK set to the value of EFN. Processing continues at SIOA9.

3.1.7.11.1 SOIA5 - At label SOIA5, the insertion of an emitter track into an existing azimuth chain is performed. The current value of the azimuth link (AZLNK) of the AZ entry shall be saved temporarily (TEMP). AZLNK shall then be set to the value of EFN and the backward azimuth link (EFBLNK) of the EF entry set to the value of TEMP. TEMP (which is an emitter file number) shall be converted into an EF entry address by calling SOGET (with the value of TEMP in the A-register). Then the forward azimuth link (EFLNK) of this second EF entry shall be set to the value of the emitter file number computed at the start of SOIAL. Processing continues at SOIA9.

3.1.7.11.2 SOIA9 - Control shall be returned to the calling routine.

3.1.8 Multi-Frequency Flag Message Processing (SOMFF)

SOMFF shall be called by SODR to process Sorter Multi-Frequency Flag messages (Op-Code = X'92'). SOGET shall immediately be called to convert the Sorter File Number (which was passed in the A-register) into the address of an Emitter Track File entry (EF entry). This pointer shall be returned by SOGET in the B-register. The Multi-Frequency bit (EFMF) shall be retrieved from the EF entry. If EFMF is already set, control shall be sent to SONAX (see 3.1.3.2) to return control to the Sorter message driver. Otherwise, EFMF shall be set in the EF entry and control sent to SOC1Ø (see 3.1.3) to output a Sorter Instrumentation message.

3.1.9 Inactive File Alert Processing (SODEL)

SODEL shall be called by SODR to process Sorter Inactive File Alerts (Op-Code = X'87'). The Sorter File Number (SFN) which is passed to SODEL in the A-register shall be saved as SOSFN in both SOUPM and SOSDF. SOUPM shall be a message buffer which has the Update Message to RMP format (see Figure 7). SOSDF shall be a message buffer which has the Delete Track File Message to Sorter format (see Figure 4). SOGET shall then be called to convert SFN into the address of an Emitter Track File entry (EF entry). This pointer shall be returned by SOGET in the B-register. The platform link field shall be retrieved from the EF entry and the emitter checked for platform linkage. If the emitter is not linked, processing continues at SOD5Ø. If it is linked, the Delete Link Processing routine (DUMMY) shall be called, with the pointer to the EF entry in the B-register.

3.1.9.1 SOD5Ø

SOIE, the Initialize EF entry subroutine shall be called to remove the old data from the EF entry. The EXEC shall then be called to output the Delete Track File Message stored in SOSDF and called again to output the deletion information stored in SOUPM to the RMP. Control shall then be transferred to label SONAX (see 3.1.3.2) to return control back to the driver (No analysis return).

3.1.9.2 Delete Link Processing

This shall be a dummy routine in the priority 1 SC operational software.

3.1.9.3 Initialize EF Entry (SOIE)

SOIE shall be called to restore an EF entry to its initial (inactive) state. This routine shall receive as input a pointer to the EF entry in the B-register. The Emitter File Number (EFN) shall be calculated from the EF entry address by subtracting the EF table base address and dividing the

difference by 16. Then all 16 words of the EF entry shall be set to zero and the following sequence executed:

1. The scan period (EFSPRD) shall be set to X'FF'.
2. The azimuth quality factor (EFQAZ) shall be set to 1.
3. The synthetic value of pulse offset between correlated pulse trains (EFOSET) shall be set to 7.
4. The PRI stagger level (EFSTAG) shall be set to 1.
5. The forward azimuth link (EFLNK) shall be set to EFN.
6. The backward azimuth link (EBLNK) shall be set to EFN.
7. The agile link (EFALNK) shall be set to EFN.
8. The correlated link (EFCLNK) shall be set to EFN.
9. The mode link (EMLNK) shall be set to EFN.
10. The scan state indicator (EFSIND) shall be set to 1.
11. The platform link (EFPLNK) shall be set to EFN.

Control shall then be returned to the calling routine.

3.1.10 New Emitter Processing 1 (SONE1)

SONE1 shall perform the following functions:

(a) Calculate the address (EFP) of the ETF entry to be activated by the NE Alert message from the signal sorter.

(b) Store the emitter parameters from the NE Alert message at the designated location in the ETF.

(c) Activate the designated ETF entry.

(d) Link the designated ETF entry to other entries at the same azimuth.

(e) Calculate the average PRI of the emitter and store the value (EFAVPI) at the designated ETF location.

3.1.10 -continued-

(f) Determine the validity of the EFAVPI value.

(g) Pass an analysis request message for deinterleaving to the Sorter message driver (SODR).

The logic flow for New Emitter Processing 1 shall be as shown in section 3.2.10.

A subroutine call shall be made to NE Processing 1 (SONE1) with a pointer to the first word of the NE Alert message contained in the X-register. The A-register shall contain the emitter file number (EFN) in the least significant byte.

SONE1 shall immediately call subroutine SOGET which shall compute the address of the emitter track file (ETF) entry and shall return it in the B-register as EFP. SONE1 shall then set the EFACT bit to indicate that the ETF entry is active.

The emitter parameters in the NE Alert message shall be stored in the ETF entry pointed to by the B-register. SONE1 shall store TA, TCW, TQPRI, TQPW, TQF, and TQAZ directly into EFA, EFCW, EFQPRI, EFQPW, EFQF, and EFQAZ respectively. SONE1 shall call SOLA to store TAZ and TPAMP into EFAZ and EFPAMP respectively. SONE1 shall call SOLB to store TPW, TFREQ, and AVTPRI = $\frac{\text{TPRIA} + \text{TPRIB}}{2}$ into EFPW, EFFREQ, and EFAVPI respectively.

The azimuth links EFLNK and EFBLNK shall be checked to link the designated ETF entry to other entries at the same azimuth (if any) by calling SOIAL.

PRI Test 1 (SOPT1) shall be called by SONE1 to assess the validity of the average PRI value (EFAVPI). SONE1 shall test the return from SOPT1 to determine if deinterleaving should be requested. If deinterleaving is not required, SONE1 shall reset the analysis wanted (SOAW) and the deinterleaving request (SODI) bits to zero in the analysis request message (see Figure 2 for format). If deinterleaving is required, SONE1 shall reset SOAW to zero and shall set SODI to one. (When deinterleaving is implemented, AW will be set to one at this point). In either case SONE1 shall store the EFN in the analysis request message. SONE1 shall load the address of the analysis request message into the X-register and shall return to the sorter message driver (SODR).

3.1.10.1 PRI Test 1

The logic flow for PRI Test 1 (SOPT1) shall be as shown in 3.2.10.1. A subroutine call shall be made to SOPT1 with the address of the ETF entry (EFP) in the B-register SOPT1 shall establish a local data area to store PARAM, N, and QVAL in consectutive locations. SOPT1 shall set PARAM equal to the value of EFAVPI (EFP). PARAM shall then be left shifted to place the most siginificant bit of the data field in the data field in the sign position. QVAL shall be set equal to QPRI (EFP) and M shall be set equal to 13.

SOPT1 shall call parameter quality test (SOQUT) with a pointer in the X-register to PARAM. SOQUT shall return with an indication of good quality (GDQ) contained in the A-register. The value of GDQ shall be stored in the PRI validity bit EFPIV (EFP).

GDQ shall be tested for zero (bad quality) and if zero, the return address shall be incremented by one to provide an analysis request return. If GDQ is equal to one, a normal return shall be made signifiying that a request for no analysis shall be made.

3.1.10.1.1 Parameter Quality Test - The logic flow for parameter quality test (SOQUT) shall be as shown in 3.2.10.1.1. A subroutine call shall be made to SOQUT with a pointer in the X-register to the parameter value (PARAM). M and QVAL shall be stored in consectutive locations following PARAM.

The value of PARAM shall be tested for sign. If PARAM is non-negative (sign bit is zero), then M shall be decremented by one. If M > 4, PARAM shall be left shifted by one and tested again. This operation shall determine the number of significant bits in PARAM. If M > 4, QVAL shall be tested for ≤ 2. If QVAL ≤ 2, the good quality bit (GDQ) shall be set to one. If QVAL > 2, GDQ shall be reset to zero.

If PARAM is found to be negative and M > 4, then QVAL shall be compared to M-2. If QVAL ≤ M-2, GDQ shall be set to one. If QVAL ≤ M-2, GDQ shall be reset to zero.

The value of GDQ shall be placed in the A-register and a return shall be made to the calling routine.

3.1.11 Sorter Instrumentation (SOINS)

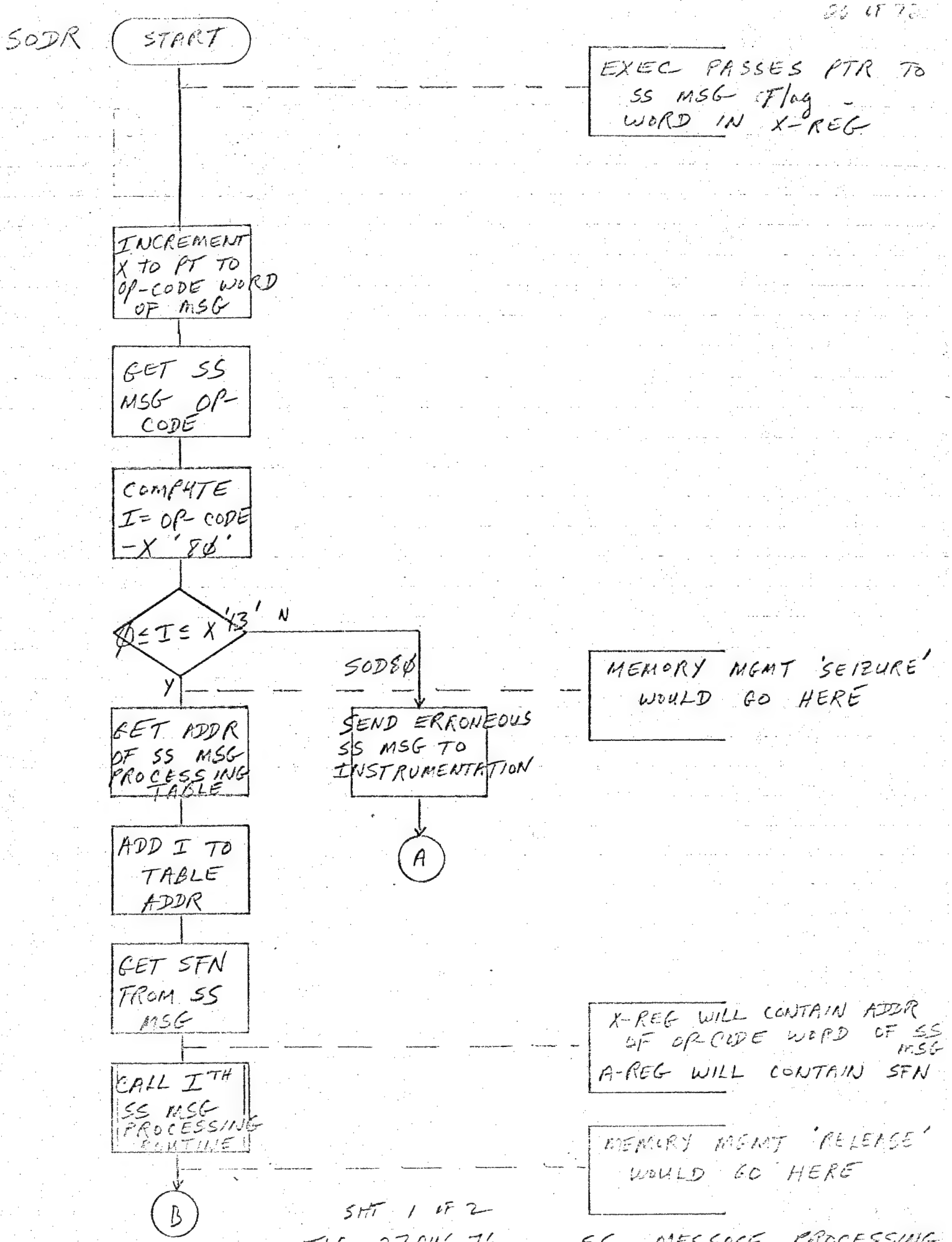
SOINS shall be called to process the following Sorter messages:

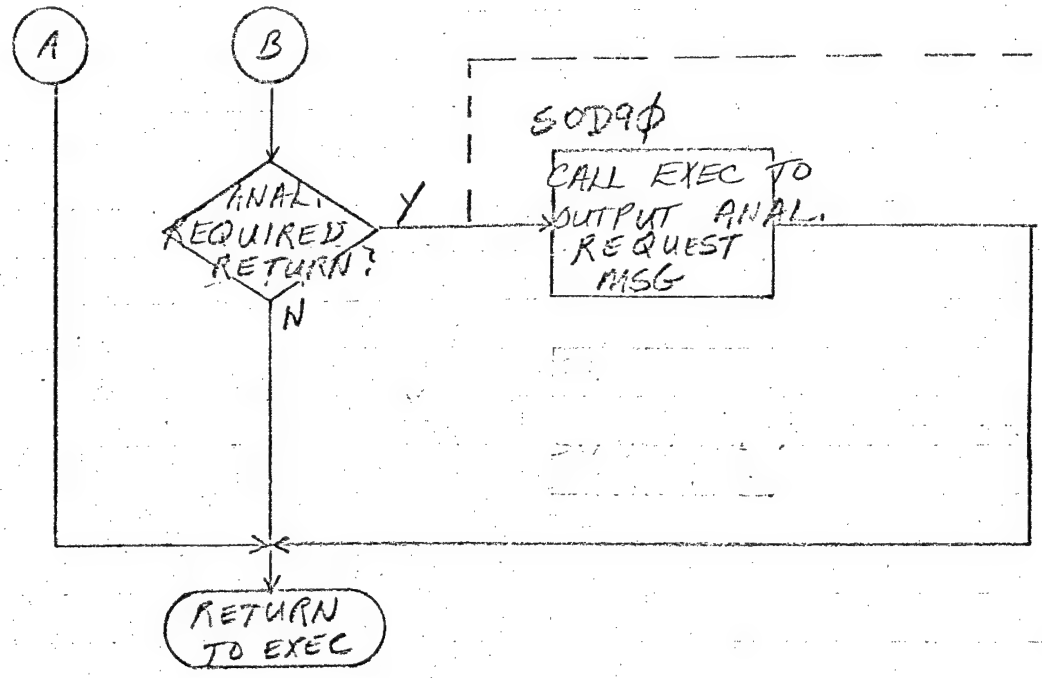
X'82' Cam File Dump
X'83' AOA Readout
X'86' Error Alert
X'8D' Bus Hung
X'8E' Watchdog Timer
X'9Ø' NPDW Message
X'91' Memory Dump
X'93' BIT Status

The name SOINS is simply a synonym for SOCFC. See 3.1.3.

3.2 SUBPROGRAM FLOW DIAGRAMS

The logic flow for all routines comprising this subprogram is shown in the following flow diagrams. The flow diagrams are labeled so as to correspond to paragraph 3.1. That is, flow diagram 3.2.9 is described in paragraph 3.1.9. Data extraction points for instrumentation are shown as comment blocks with the text "DP_____".

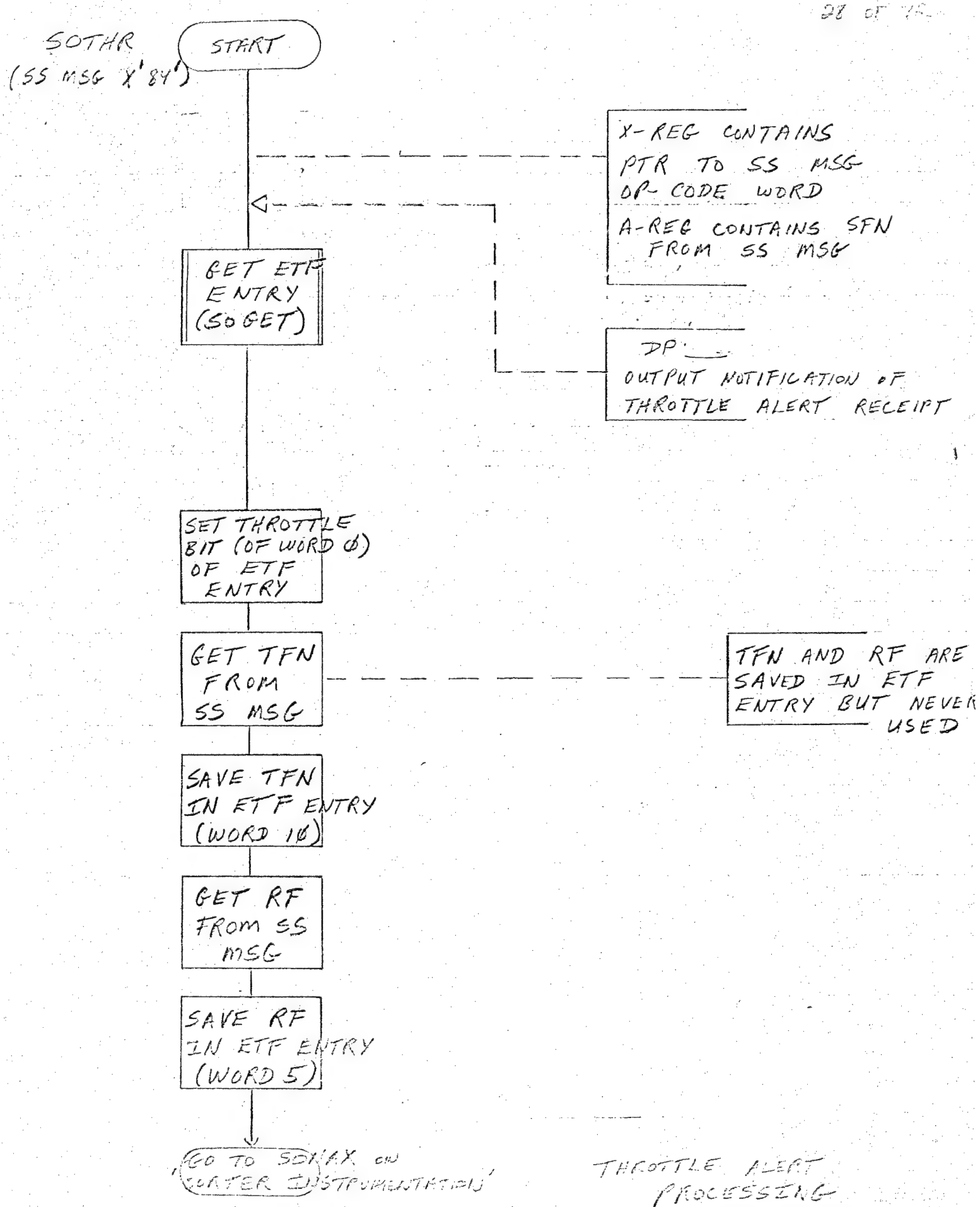

SODR
START
EXEC PASSES PTR TO SS MSG Q/q - WORD IN X-REG
INCREMENT X TO PT TO OP-CODE WORD OF MSG
GET SS MSG OP-CODE
COMPUTE I = OP-CODE - X '80'
Ø ≤ I ≤ X '13'
N
Y
SOD80
MEMORY MGMT 'SEIZURE' WOULD GO HERE
GET ADDR OF SS MSG PROCESSING TABLE
SEND ERRONEOUS SS MSG TO INSTRUMENTATION
A
ADD I TO TABLE ADDR
GET SFN FROM SS MSG
X-REG WILL CONTAIN ADDR OF OP-CODE WORD OF SS MSG
A-REG WILL CONTAIN SFN
CALL Ith SS MSG PROCESSING ROUTINE
MEMORY MGMT 'RELEASE' WOULD GO HERE
B



IF ANAL REQUEST MSG IS TO BE OUTPUT, SS MSG PROCESSING ROUTINE MUST RETURN PTR TO MSG IN X-RG

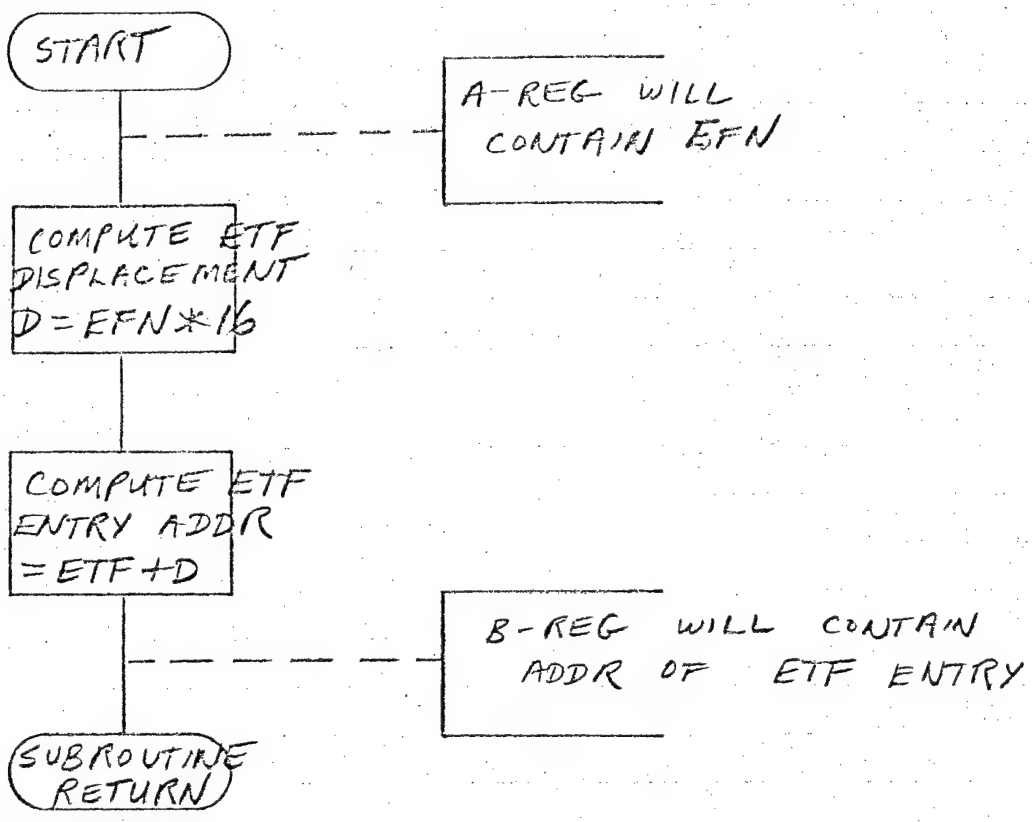
SHT 2 OF 2 SS MESSAGE PROCESSING

THC 27 AUG 76



THROTTLE ALERT PROCESSING
TAC 19 AUG 76

SOGET



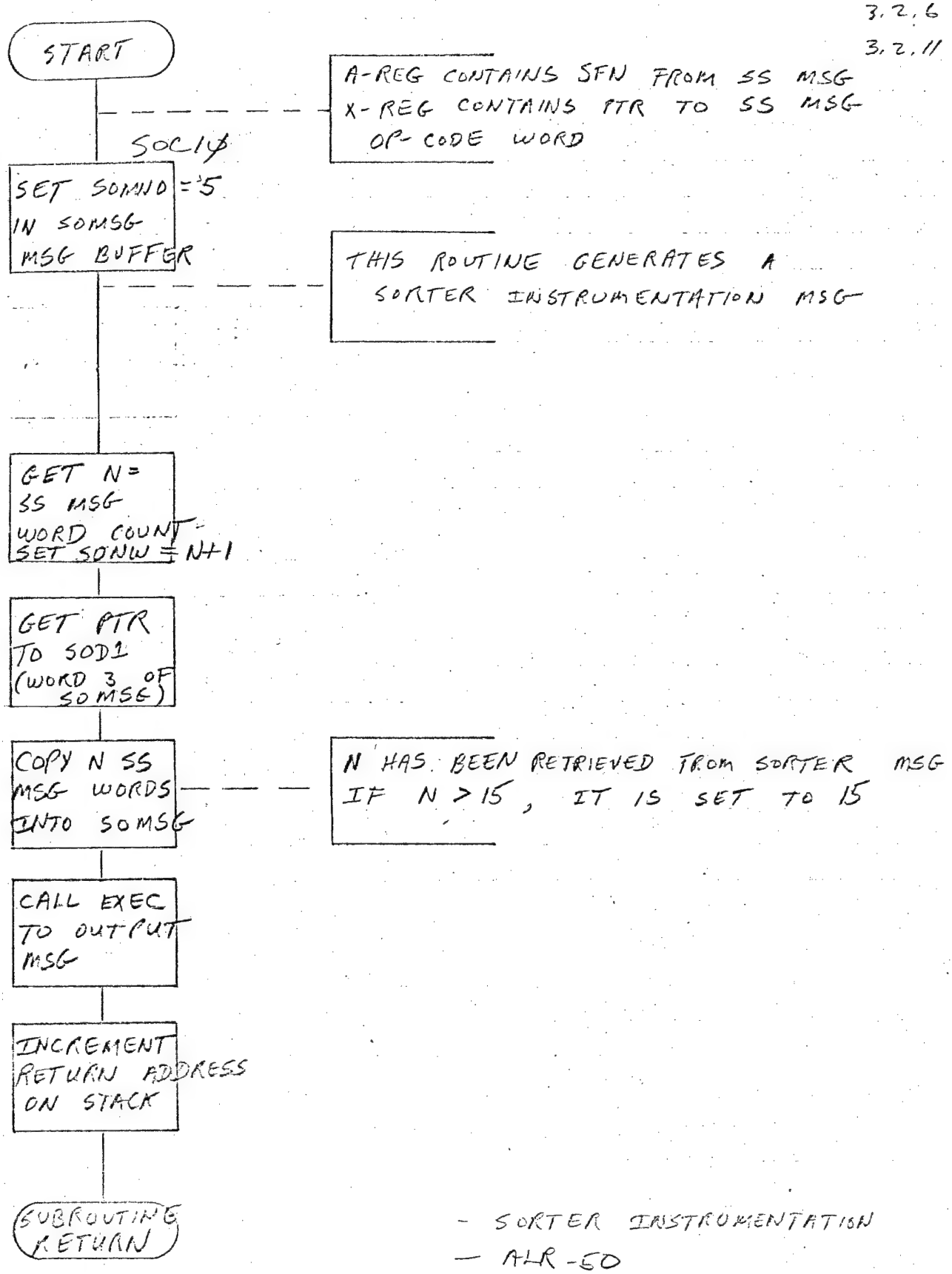
GET ETF ENTRY ADDR

TLC 24 AUG 76

3.2.3
3.2.5
3.2.6
3.2.11

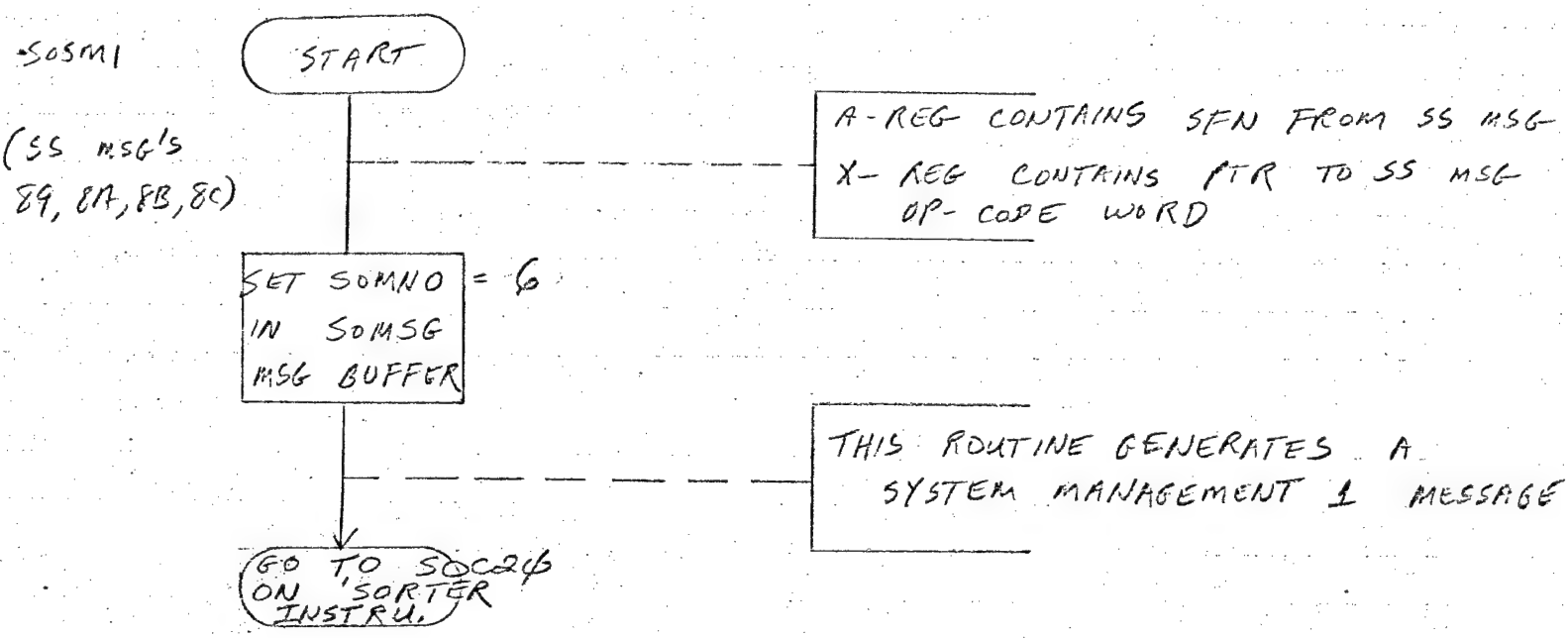
SOCFC
SOLP
SOALR
SOINS

(SS MSG's 82, 83, 85, 86, 88, 8D 8E, 8F, 9Ø 91, 93)



- SORTER INSTRUMENTATION
— ALR-EO
- LONG PULSE
— CONFIRM FILE CREATION

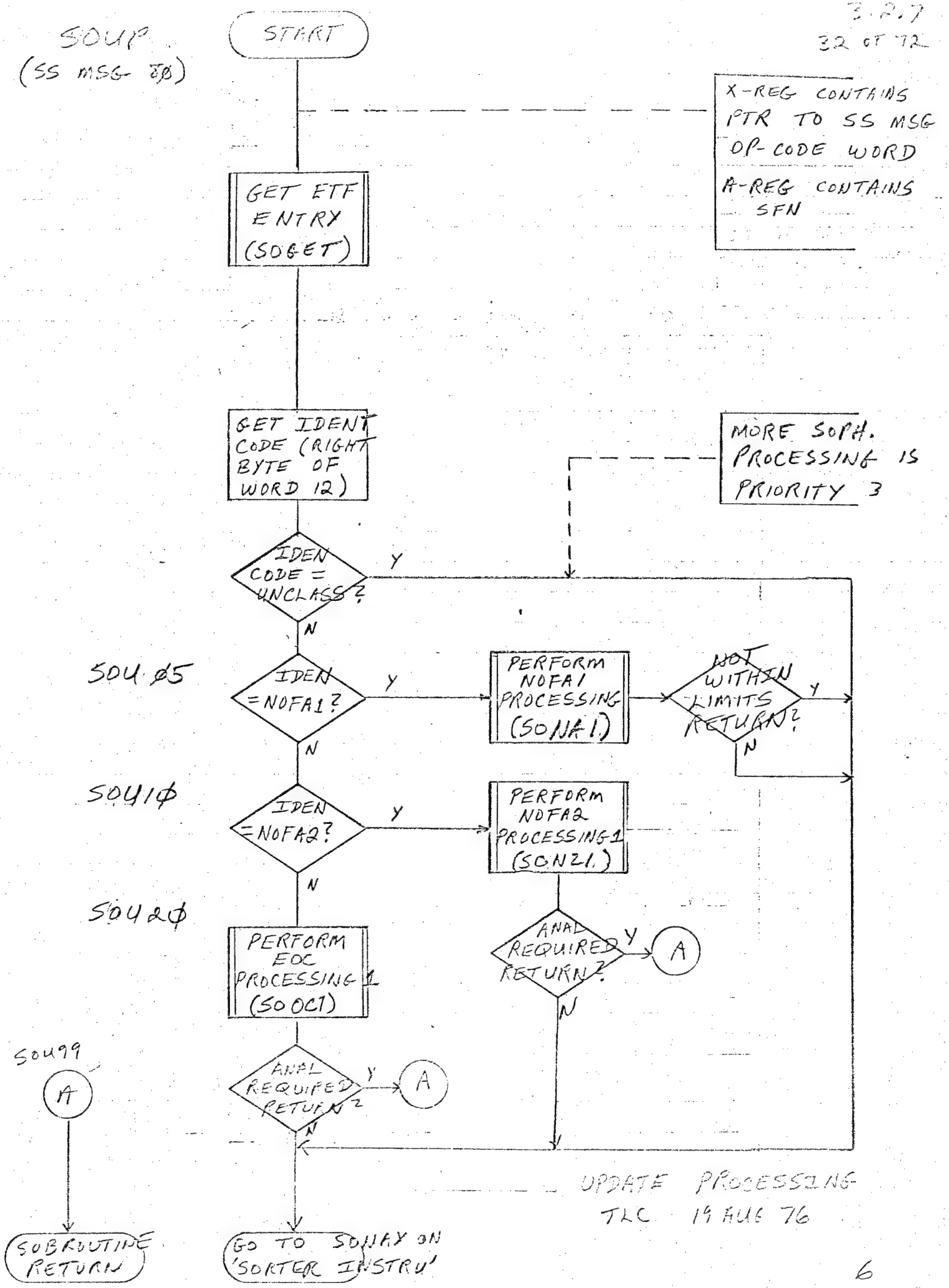
TLC 2 SEP 76



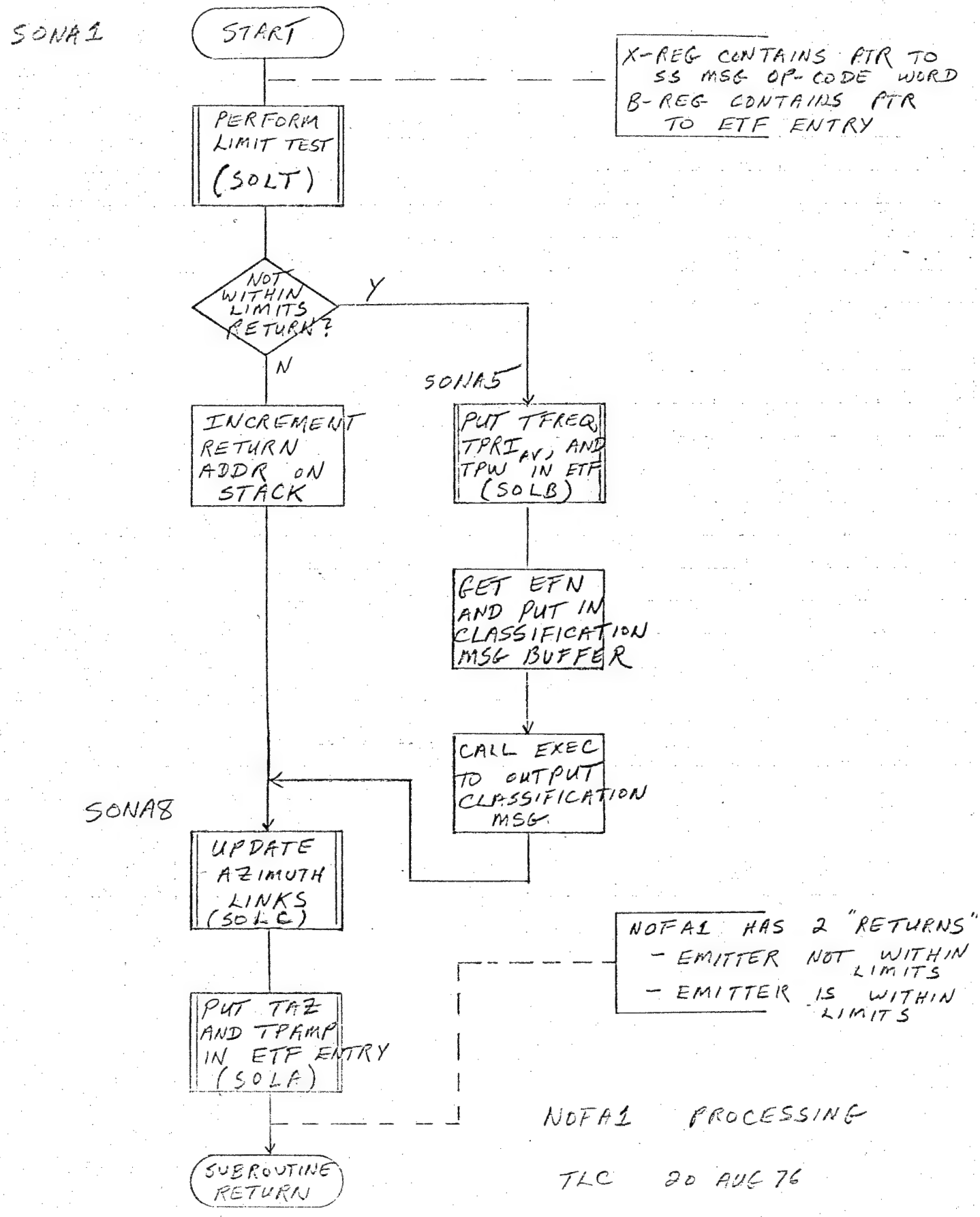
SYSTEM MANAGEMENT 1

TLC 2 SEP 76

SOUP
(SS MSG EP)

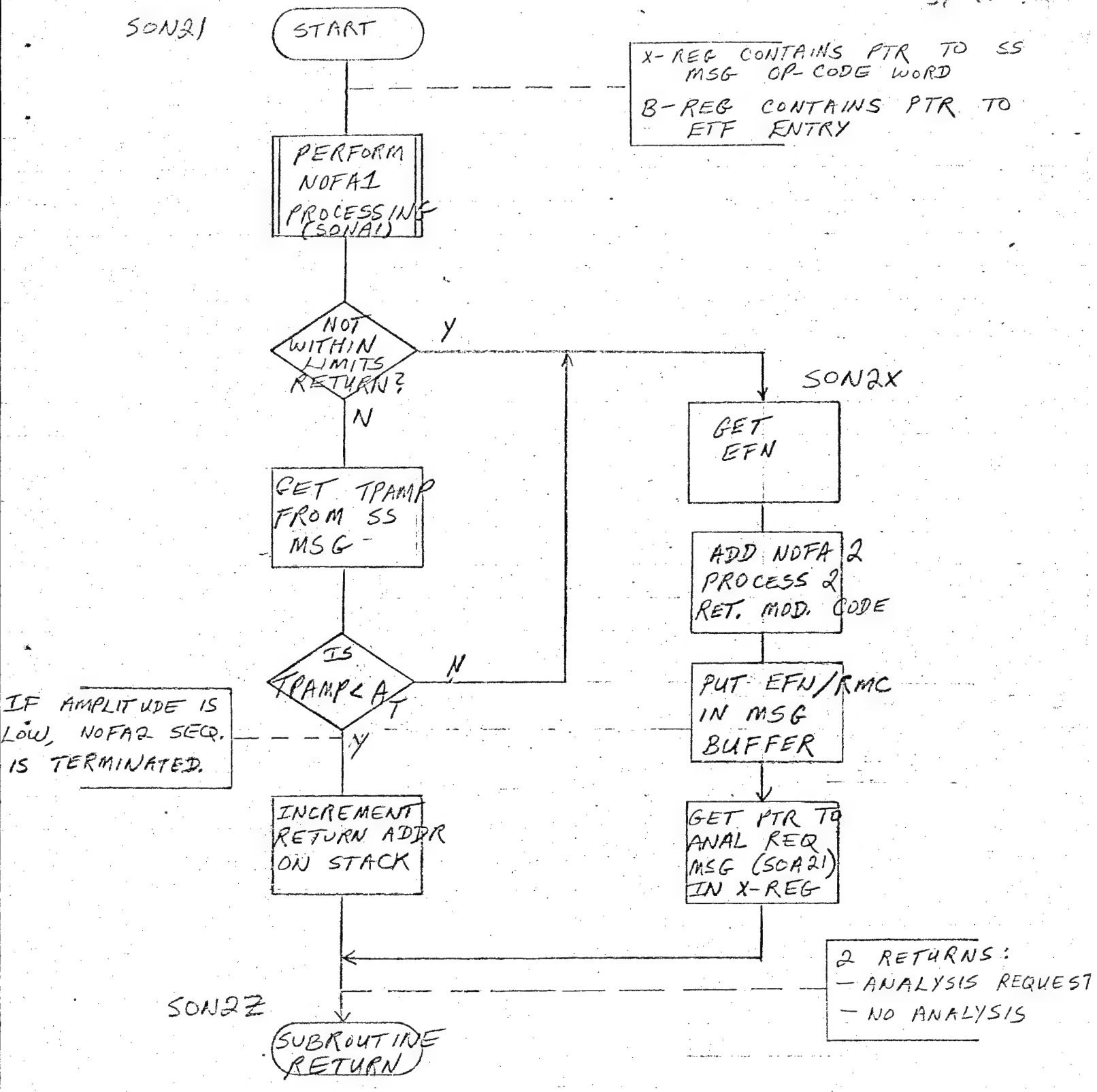


UPDATE PROCESSING
TLC 19 AUG 76



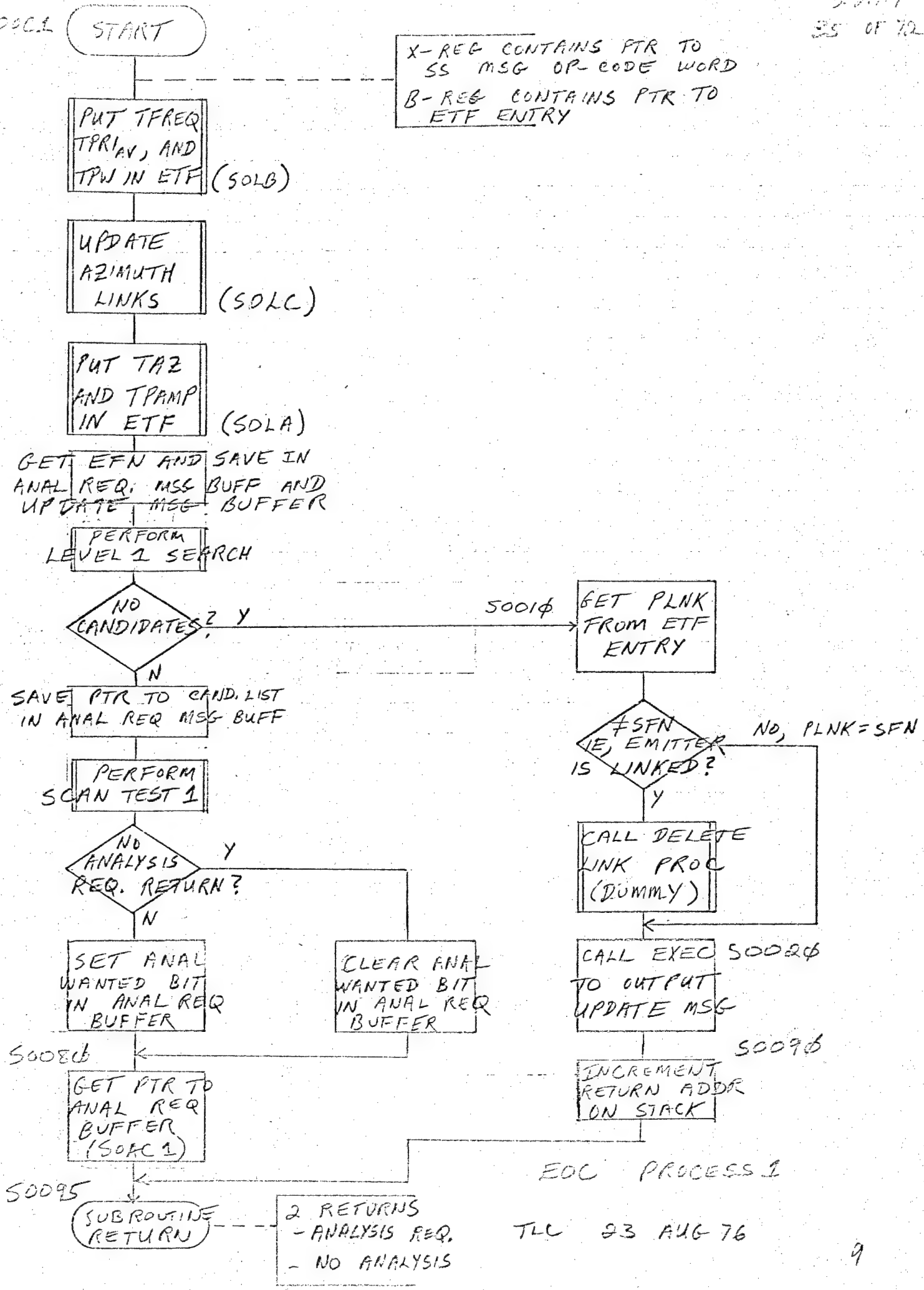
NOFA1 PROCESSING

TLC 20 AUG 76

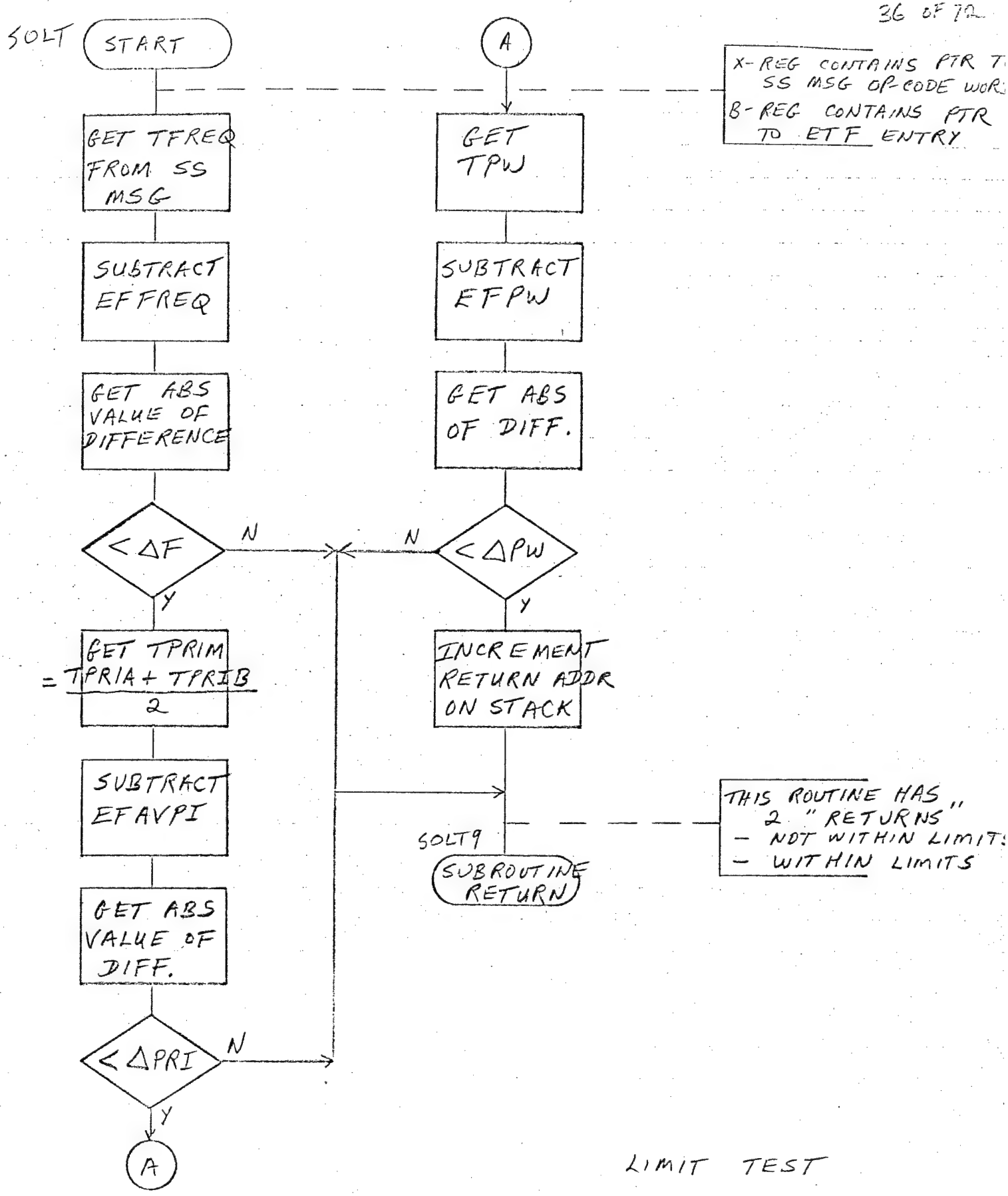


NOFA 2
PROCESS 1

TLC 83 AUG 76


S00C1
START
X-REG CONTAINS PTR TO SS MSG OR CODE WORD
B-REG CONTAINS PTR TO ETF ENTRY
PUT TFREQ TPRI$_{AV}$, AND TPW IN ETF (S0LB)
UPDATE AZIMUTH LINKS (S0LC)
PUT TAZ AND TPAMP IN ETF (S0LA)
GET EFN AND SAVE IN ANAL REQ. MSG BUFF AND UPDATE MSG BUFFER
PERFORM LEVEL 1 SEARCH
NO CANDIDATES? Y
N
S001Ø
GET PLNK FROM ETF ENTRY
SAVE PTR TO CAND. LIST IN ANAL REQ MSG BUFF
PERFORM SCAN TEST 1
NO ANALYSIS REQ. RETURN? Y
N
≠SFN (IE, EMITTER IS LINKED? NO, PLNK=SFN
Y
CALL DELETE LINK PROC (DUMMY)
SET ANAL WANTED BIT IN ANAL REQ BUFFER
CLEAR ANAL WANTED BIT IN ANAL REQ BUFFER
CALL EXEC S002Ø TO OUTPUT UPDATE MSG
S008Ø
S009Ø
GET PTR TO ANAL REQ BUFFER (S0AC1)
INCREMENT RETURN ADDR ON STACK
EOC PROCESS 1
S0095
SUBROUTINE RETURN
2 RETURNS
- ANALYSIS REQ.
- NO ANALYSIS

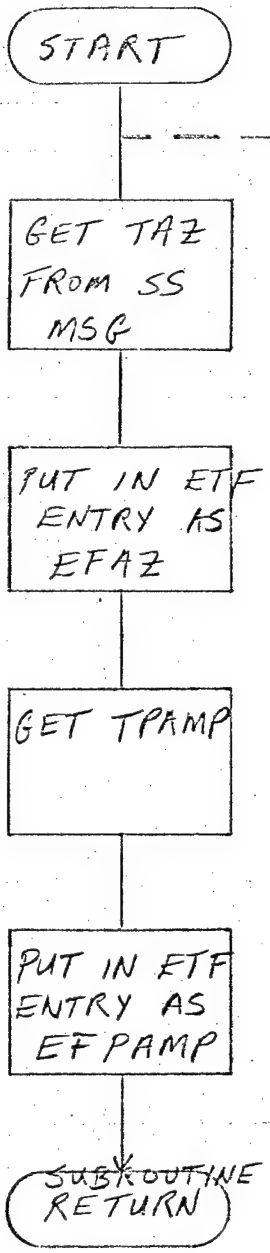
TLC 23 AUG 76



LIMIT TEST

TLC : 23 AUG 76

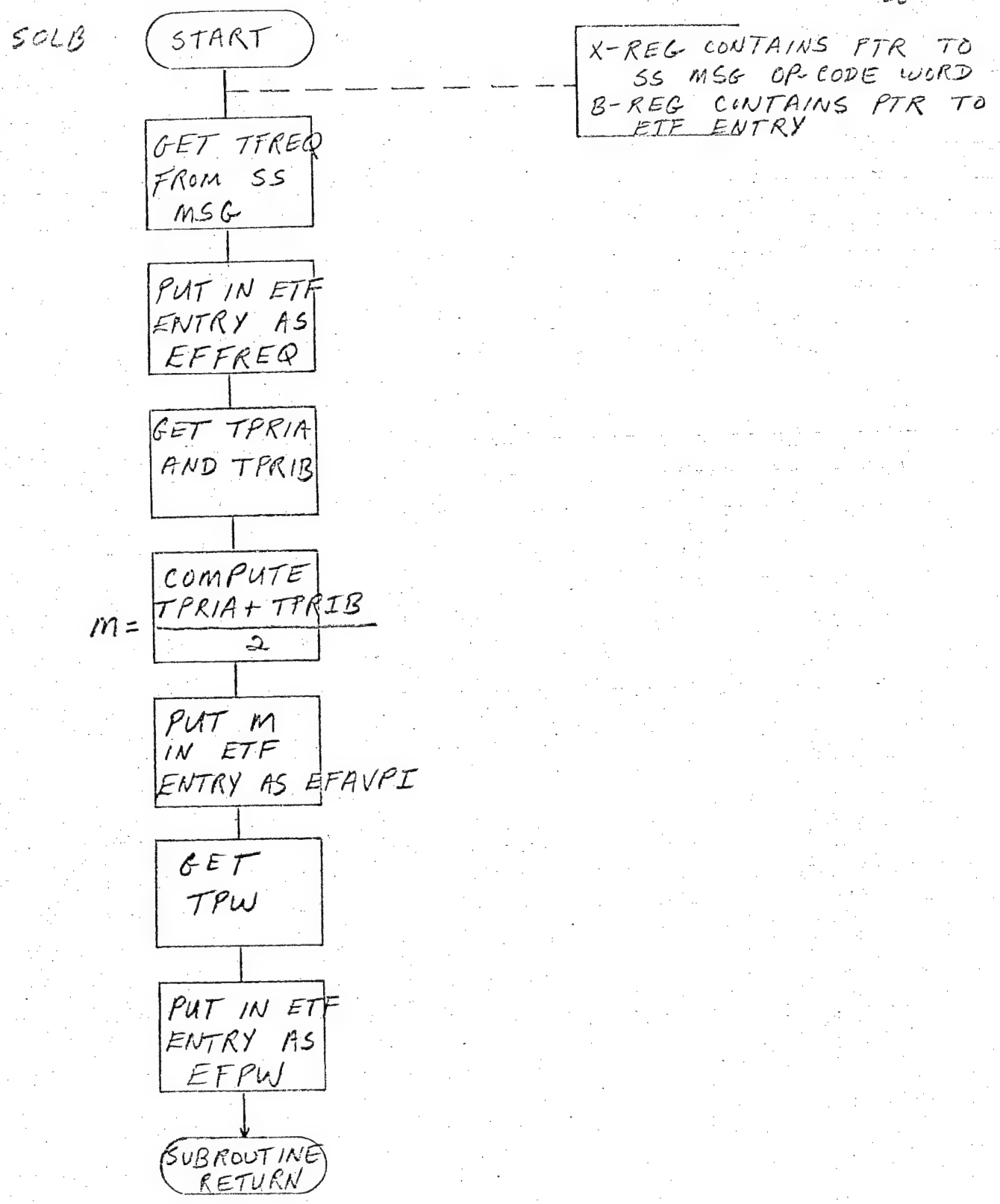
SOLA



X-REG CONTAINS PTR TO SS MSG OP-CODE WORD
B-REG CONTAINS PTR TO ETF ENTRY

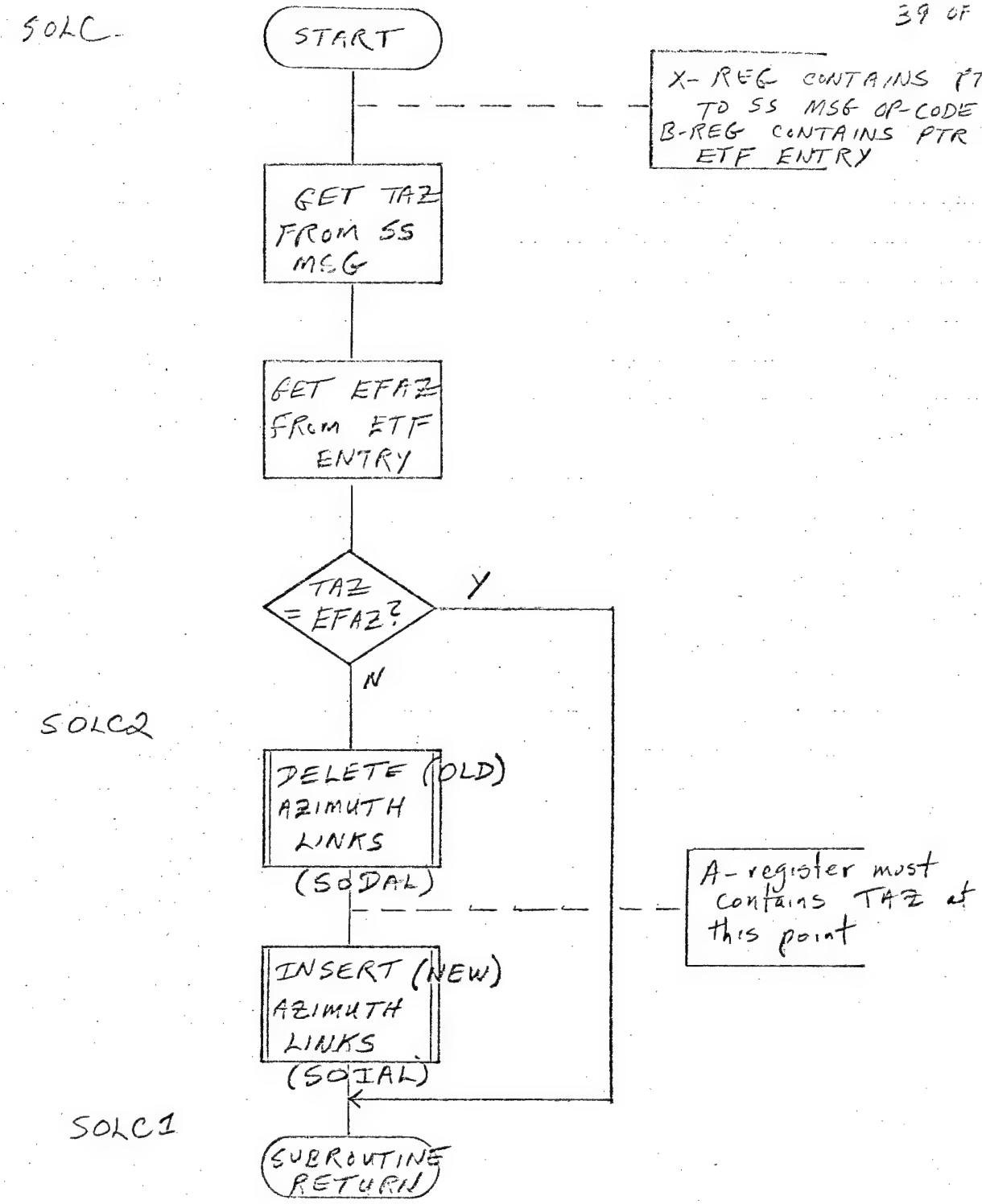
LOAD TAZ AND TPAMP INTO ETF ENTRY

TLC 23 AUG 76



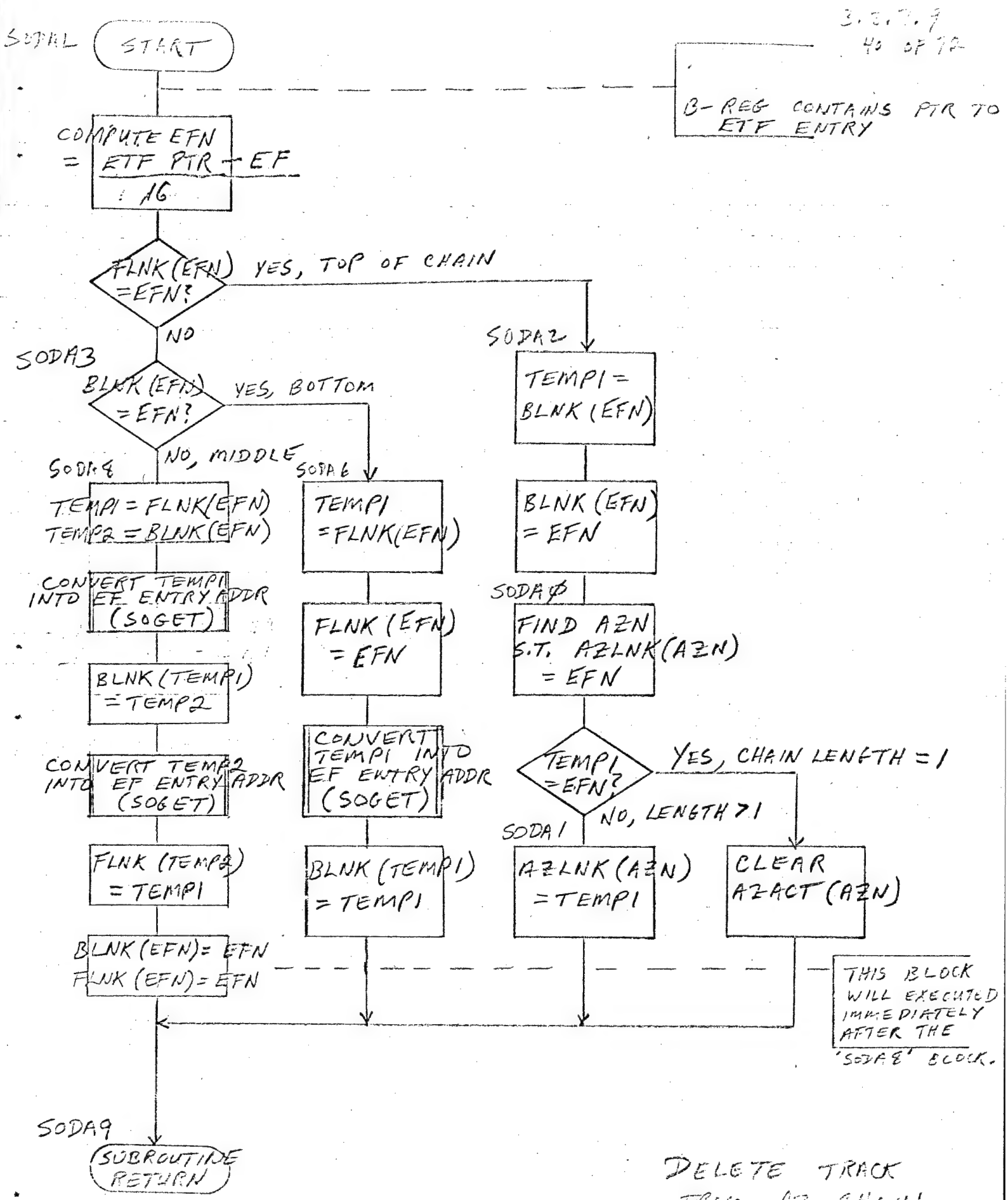
LOAD TFREQ, AVE. TPRI, AND TPW INTO ETF ENTR

TLC 23AUG 76



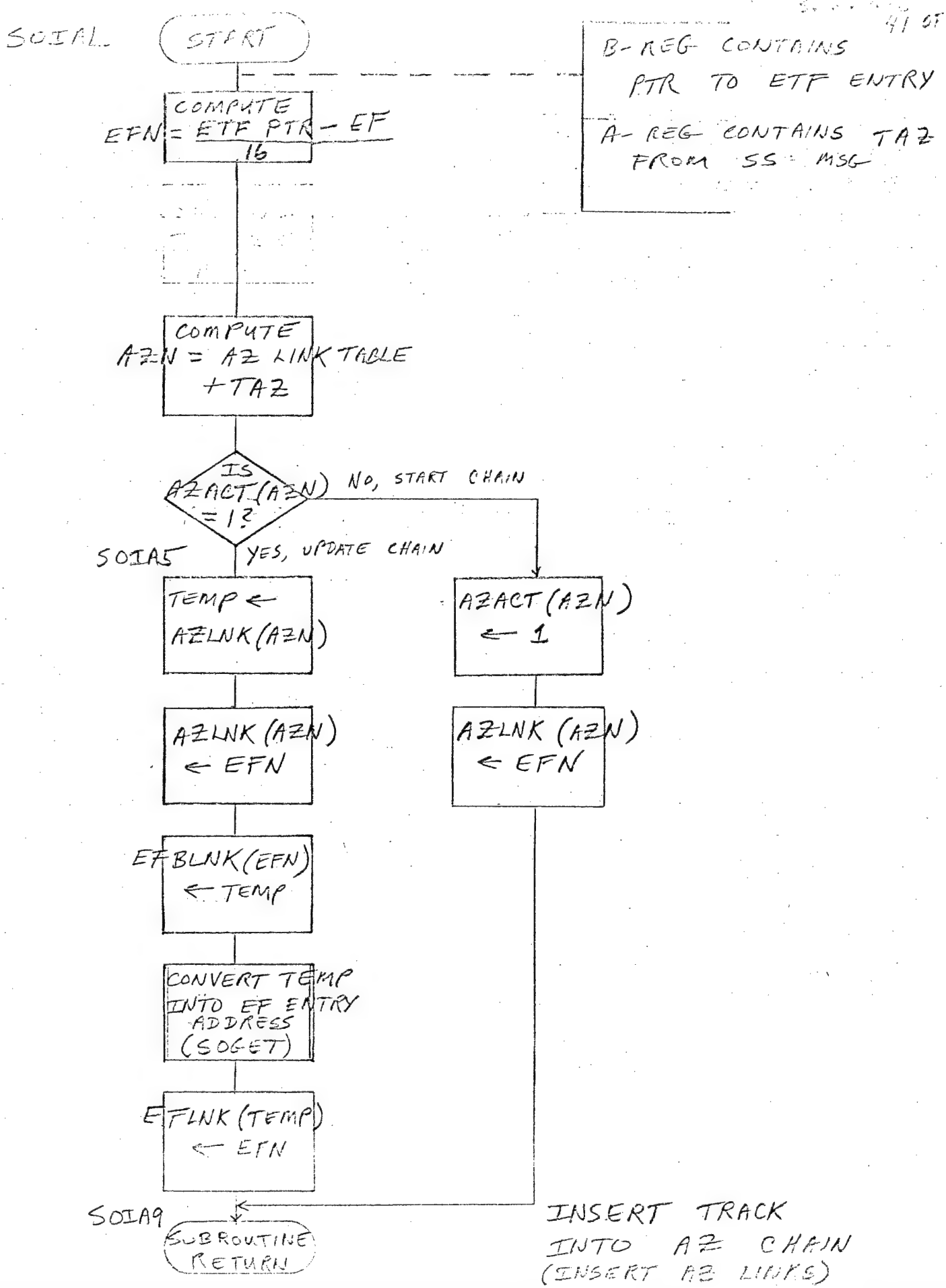
UPDATE AZ LINKS

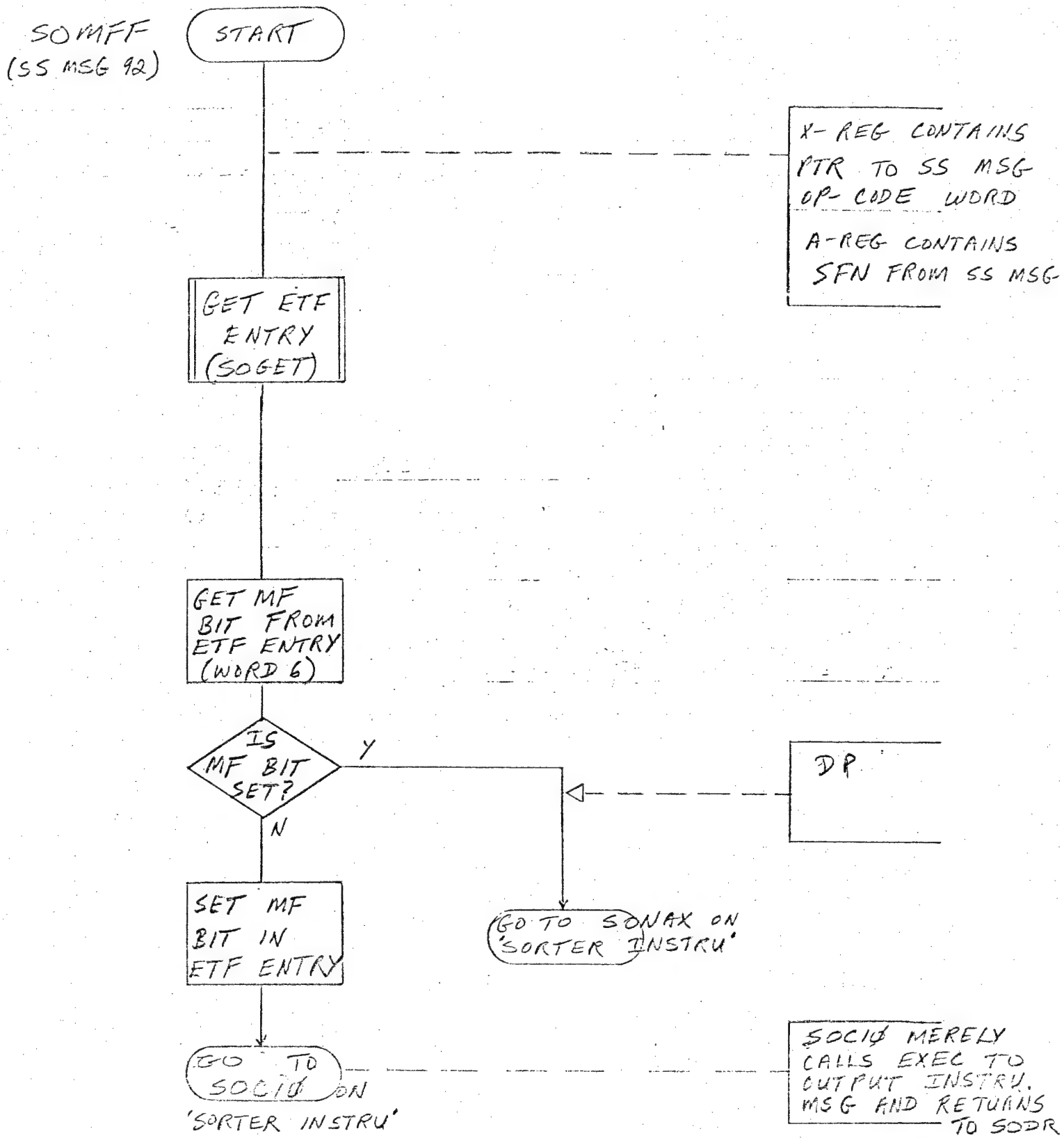
TLC 23 AUG 76



DELETE TRACK FROM AZ CHAIN (DELETE AZ LINKS)

TLC 23 AUG 76

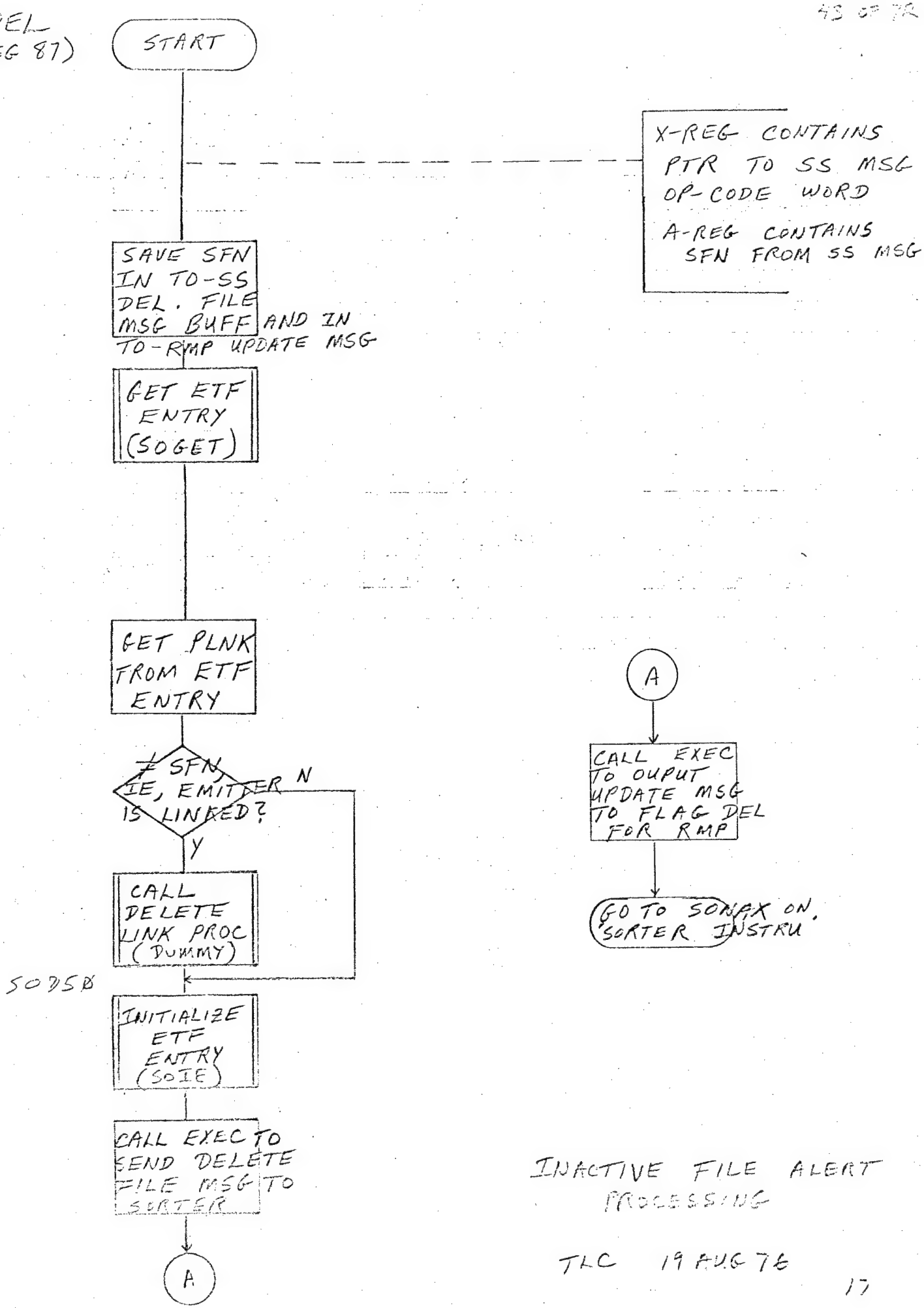

SOIAL
START
B-REG CONTAINS PTR TO ETF ENTRY
A-REG CONTAINS TAZ FROM SS MSG
COMPUTE EFN = ETF PTR − EF / 16
COMPUTE AZN = AZ LINK TABLE + TAZ
IS AZACT(AZN) = 1?
NO, START CHAIN
SOIA5
YES, UPDATE CHAIN
TEMP ← AZLNK(AZN)
AZACT(AZN) ← 1
AZLNK(AZN) ← EFN
AZLNK(AZN) ← EFN
EFBLNK(EFN) ← TEMP
CONVERT TEMP INTO EF ENTRY ADDRESS (SOGET)
EFLNK(TEMP) ← EFN
SOIA9
SUBROUTINE RETURN
INSERT TRACK INTO AZ CHAIN (INSERT AZ LINKS)
TLC 23 AUG 76



MFF MESSAGE PROCESSING

TLC 19AUG 76

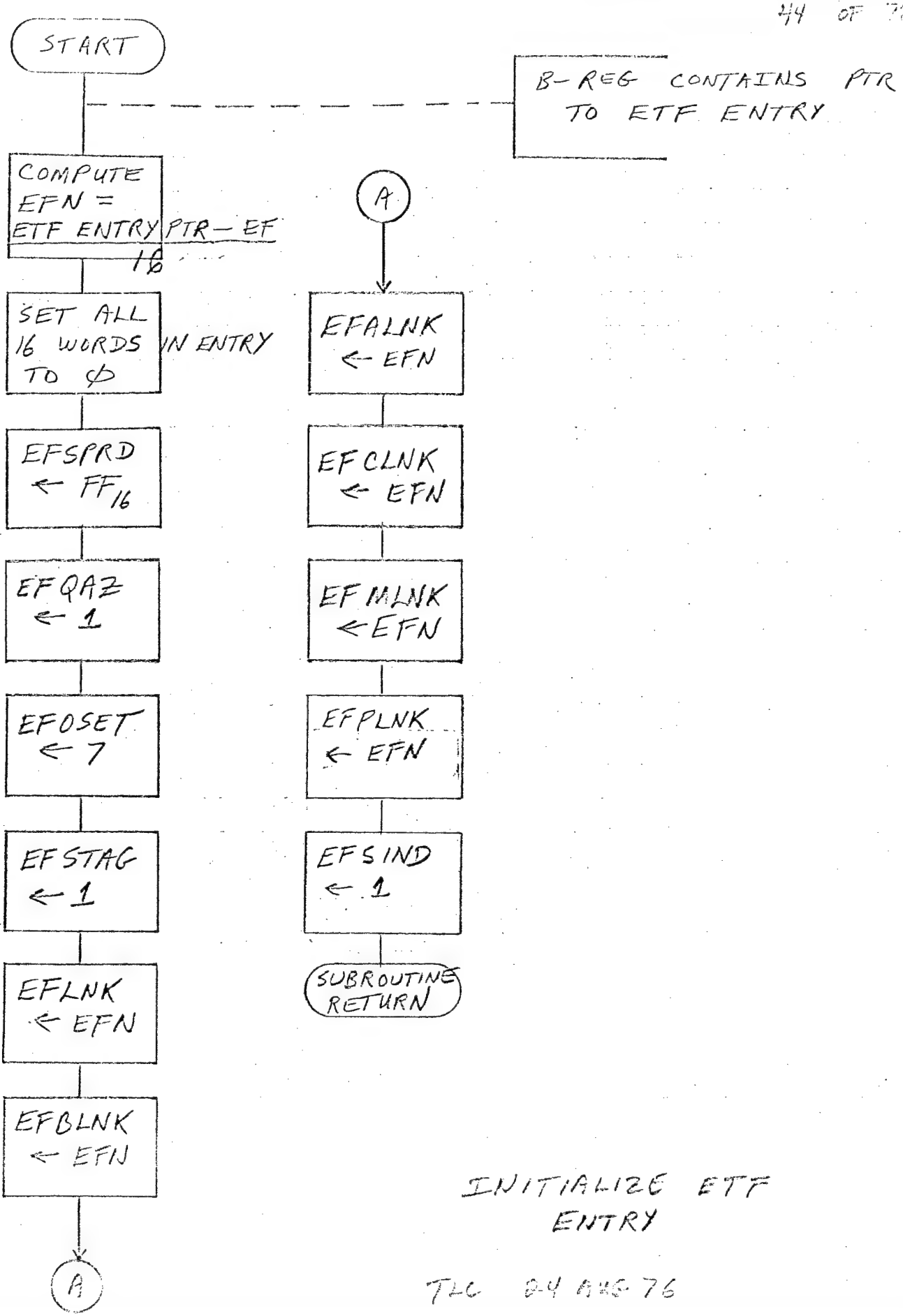
SODEL
(SS MSG 87)



INACTIVE FILE ALERT PROCESSING

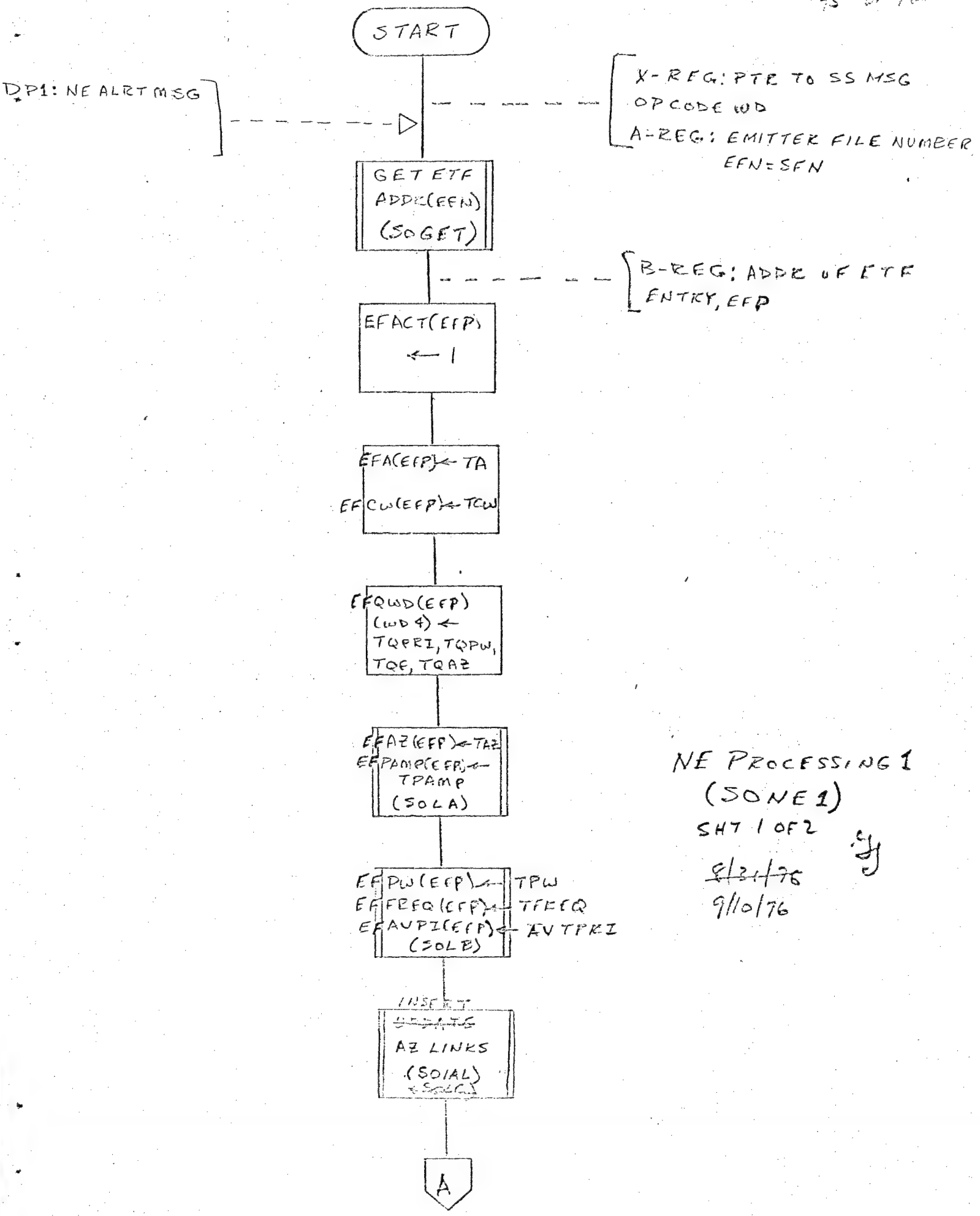
TLC 19 AUG 76

SOIE



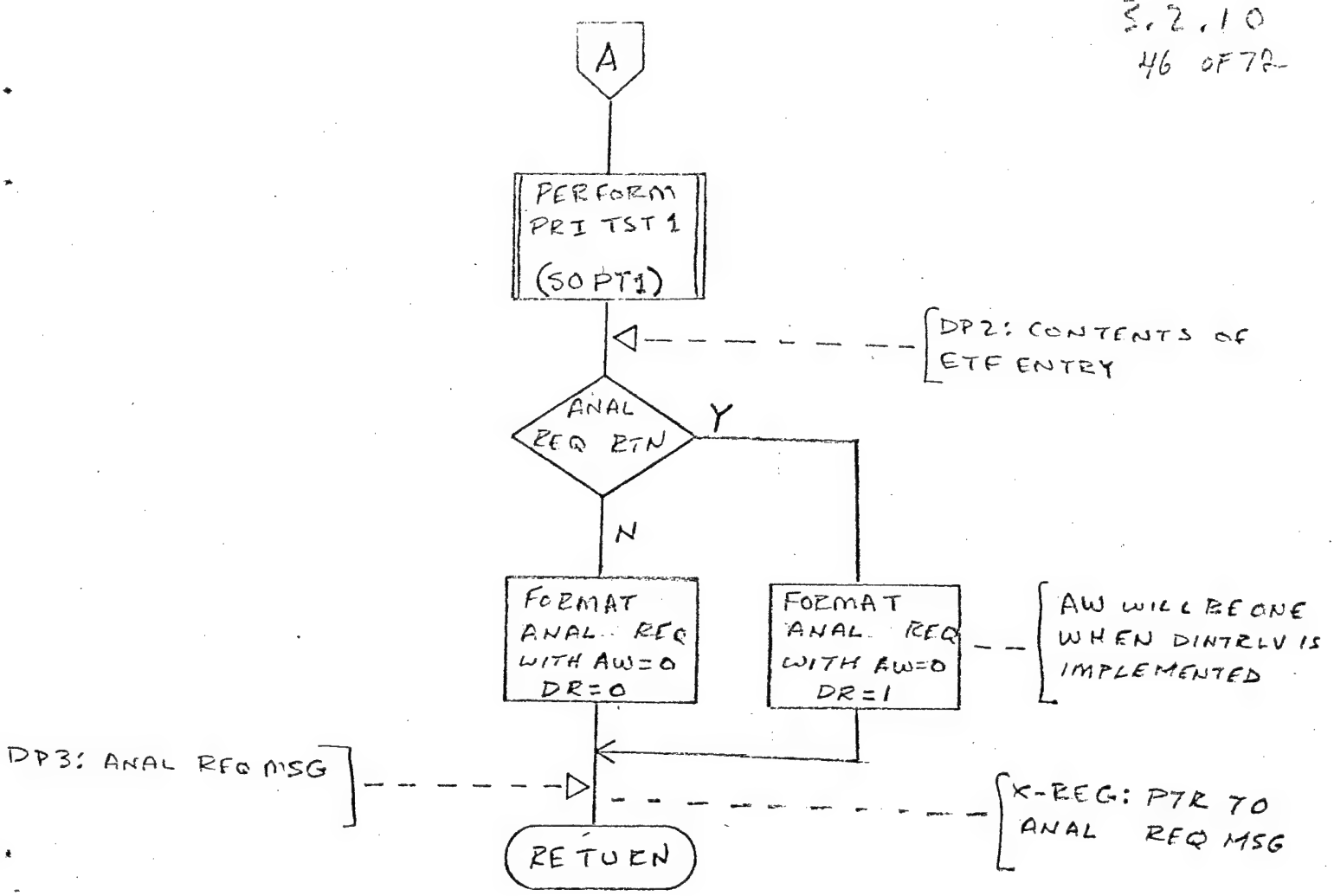
INITIALIZE ETF ENTRY

TLC 04 AUG 76

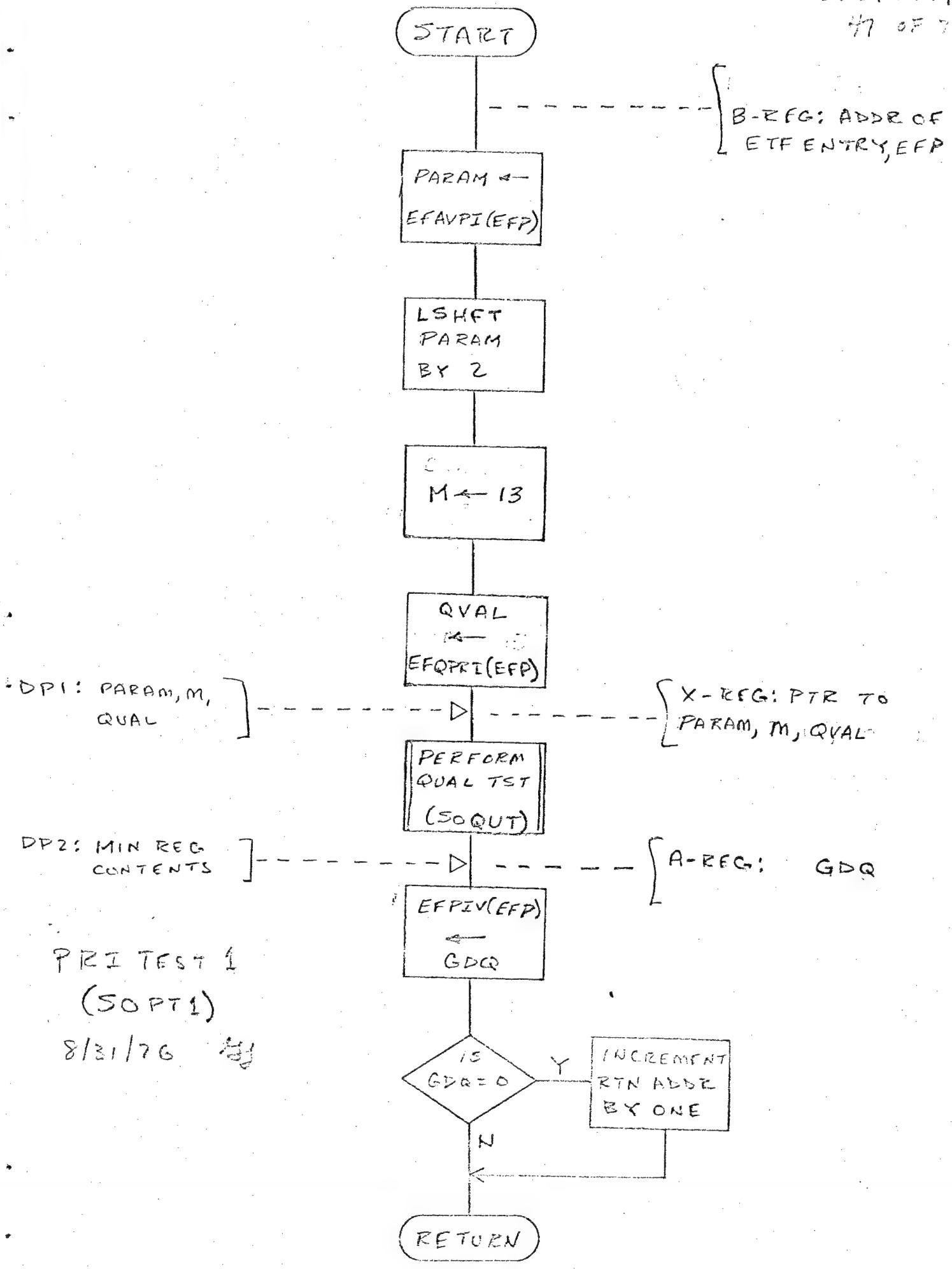


NE PROCESSING 1
(SONE1)
SHT 1 OF 2

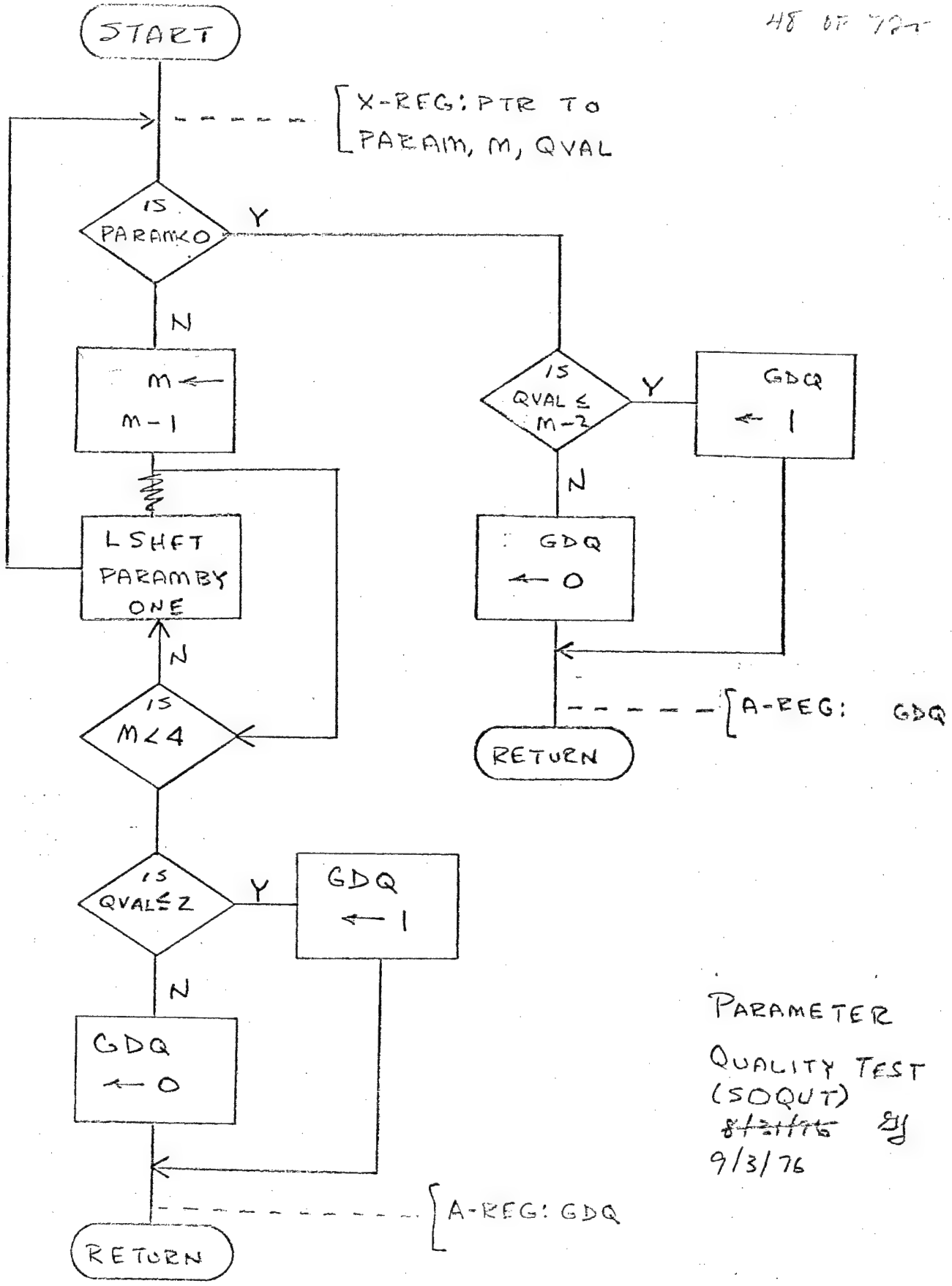
~~8/31/76~~
9/10/76



NE PROCESSING 1
(SONE1)
SHT 2 OF 2
8/31/76



PRI TEST 1
(SOPT1)
8/31/76



PARAMETER QUALITY TEST (SOQUT)
~~8/2/76~~ BJ
9/3/76

3.3 COMPUTER SUBPROGRAM ENVIRONMENT

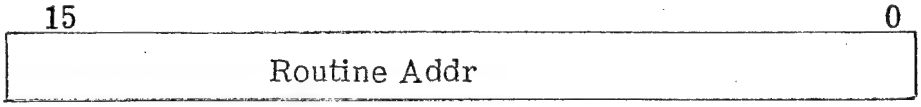
3.3.1 Tables

3.3.1.1 Sorter Message Driver Tables - Sorter Message Processing Table (SOMPT).

Purpose and Type - Fixed Length Table containing the addresses of the subroutines called to process an Sorter-to-SC message.

Size and Indexing Procedure - 2Ø entries of one 16-bit word. All entries shall be referenced by indexed displacement from the start of the table.

Entry Format -



Field	Description	Units	LSB
Routine Addr	Address of an Sorter Message Processing Routine	N/A	N/A

3.3.2 Variables

3.3.2.1 Sorter Message Driver Variables -

None.

3.3.2.2 Throttle Alert Processing Variables -

None.

3.3.2.3 Confirm File Creation Processing Variables -

SOCFC shall generate a Sorter Instrumentation message. The message data shall be stored in SOMSG. The format of SOMSG is shown in Figure 5.

3.3.2.4 System Management 1 Variables -

SOSM1 shall generate a System Management 1 message. The message data shall be stored in SOMSG. The format of SOMSG is shown in Figure 6.

3.3.2.5 Long Pulse Processing Variables -

SOLP shall generate a Sorter Instrumentation message (SOMSG). See Figure 5.

3.3.2.6 ALR-50 Processing Variables -

SOALR shall generate a Sorter Instrumentation message (SOMSG). See Figure 5.

3.3.2.7 Update Processing Variables -

3.3.2.7.1 Classification Message Buffer - NOFA1 processing shall conditionally generate a Classification Message. The message data shall be stored in SOCLM. The format of SOCLM is shown in Figure 3.

3.3.2.7.2 NOFA2 Process 1 Analysis Request Message Buffer - SON21 shall conditionally generate an Analysis Request message. The message data shall be stored in SOA21 (with SORMC=3). The format of SOA21 is shown in Figure 2.

3.3.2.7.3 EOC Process 1 Analysis Request Message Buffer - SOOC1 shall conditionally generate an Analysis Request message. The message data shall be stored in SOAC1 (with SORMC=5). The format of SOAC1 is shown in Figure 2.

3.3.2.7.4 EOC Process 1 Update Message Buffer - EOC1 shall conditionally generate an Update message. The message data shall be stored in SOUPM. The format of SOUPM is shown in Figure 7.

3.3.2.8 MFF Processing Variables

None.

3.3.2.9 Deletion Processing Variables

3.3.2.9.1 Update Message - SODEL shall generate an Update message. The update message shall be used to notify the Resource Management Processor of the deletion action. The message data shall be stored in SOUPM. The format is shown in Figure 7.

3.3.2.9.2 Delete Track File Message (to Sorter) - SODEL shall generate a Delete Track File Message to be sent to the Sorter in response to a Sorter Inactive File Alert. The message data shall be stored in SOSDF. The format is shown in Figure 4.

3.3.2.10 New Emitter Processing 1 Variables

The variables required by SONE1 are shown in Table 1.

TABLE 1

Variable Descriptions for NE Proc. 1

Descriptive Item	VARIABLE NAME					
	EFP	EFN	GDQ	M	PARAM.	QVAL
Purpose	Provides address of first word of ETF entry given by EFN.	Value of ETF entry.	Indicator of data quality.	One less than number of significant bits in PARAM.	Parameter which is to be tested for quality.	Quality factor associated with PARAM.
Type	Fixed point.	Fixed point.	Fixed point.	Fixed point.	Fixed point.	Fixed point.
Size	16	8	1	4	16	4
Binary Pt.	Bit 0	Bit 0	N/A	Bit 0	Bit 0	Bit 0
Max. Value	65,536*	127	1=good quality	15	65,536	15
Min. Value	0*	-128	0=bad quality	3	0	0
Initial Value	Don't care	Don't care	Don't care	Don't care	Don't care	Don't care
Static/ Dynamic	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic

* Memory Address Spectrum Will Restrict EFP

3.3.2.11 Sorter Instrumentation Variables

SOINS shall generate a Sorter Instrumentation message. The message data shall be stored in SOMSG. The format of SOMSG is shown in Figure 5.

3.3.3 Constants

There shall be no local constants associated with the Sorter Message Functional Group.

3.3.4 Flags

There shall be no local flags associated with the Sorter Message Functional Group.

3.3.5 Indices

The emitter file number (EFN) shall be an index that is used throughout the Sorter Message Functional Group. It shall be used to access an entry in the Emitter Track File (EF). EFN shall assume the following range of values:

$$\emptyset \leq EFN \leq 127$$

3.3.5.1 Sorter Message Driver Indices

Sorter Message Processing Table Index

a. Index Name. I (Not a symbolic label)

b. Purpose. The index shall be used to fetch a Sorter message processor routine address from table SOMPT. I shall assume the following range of values:

$$\emptyset \leq I \leq X'13'$$

3.3.6 Common Data Base References

The following items in the common data base shall be referenced by the routines in the Sorter Message Functional Group:

EF Emitter Track File
AZ Azimuth Link Table
CL Candidate List

3.4 INPUT/OUTPUT FORMATS

The format of all Sorter to SC (input) messages shall be as specified in the System Controller-Sorter ICD, 53959-JK-1002. Output message formats shall be as specified in the Executive CSDD. The format of instrumentation data output shall be as specified in the Data Extraction CSDD, 53959-GT-0759.

3.5 SYSTEM LIBRARY SUBROUTINES

There shall be no system library subroutines required by the Sorter Message Functional Group.

3.6 CONDITIONS FOR INITIALIZATION

This subprogram shall have unconditional entry and shall require no special initialization procedure.

3.7 SUBPROGRAM LIMITATIONS

The Sorter Message Functional Group shall have the following limitations:

1. SODR shall retrieve the op-code from the Sorter message and verify that it is a valid code. If not valid, an error alert message shall be sent to Instrumentation.

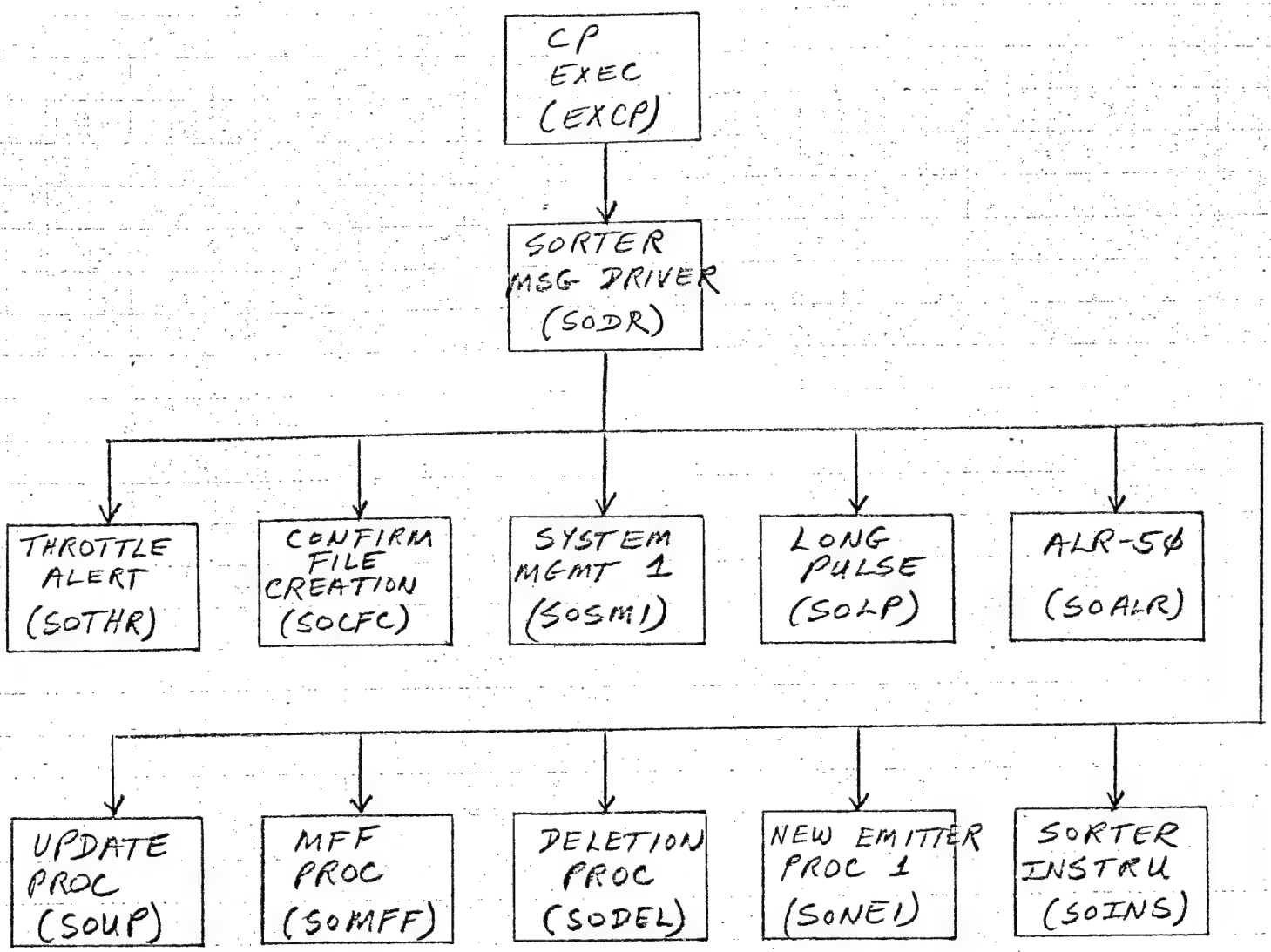
2. SOCFC, SOLP, SOALR, SOINS, and SOSM1 shall copy N words of data from the Sorter message into a message buffer (SOMSG). If N is greater than 15 (a Sorter error condition), N shall be set to 15 and processing shall continue normally.

3.7 -continued-

3. In SONE1, the PRI test 1 (SOPT1), shall assume that the PRI value contains 14 bits, right justified in one 16 bit word.

3.8 INTERFACE DESCRIPTION

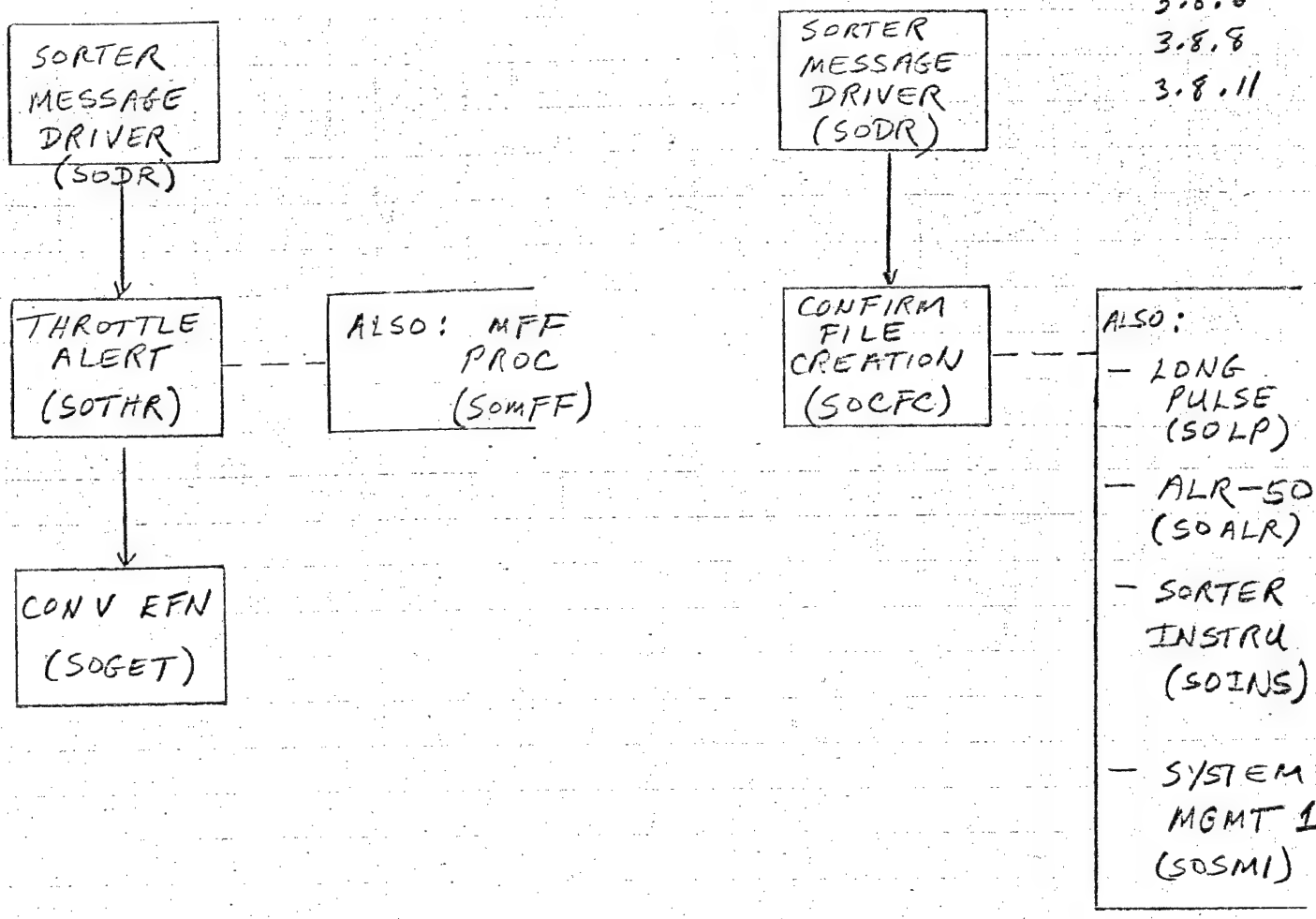
The Sorter Message Processing Driver (SODR) shall be called by the EXEC. It shall then call one of the Sorter Message Processing Routines (SOTHR, SOCFC, etc.). The routines called by each Sorter Message processor are shown in the following interface diagrams. Instrumentation shall be called as required for data extraction and is not shown on the diagrams. Calls to the Executive function to output messages are also not shown.



INTERFACE DESCRIPTION
SORTER MESSAGE DRIVER

TLC 8 SEP 76

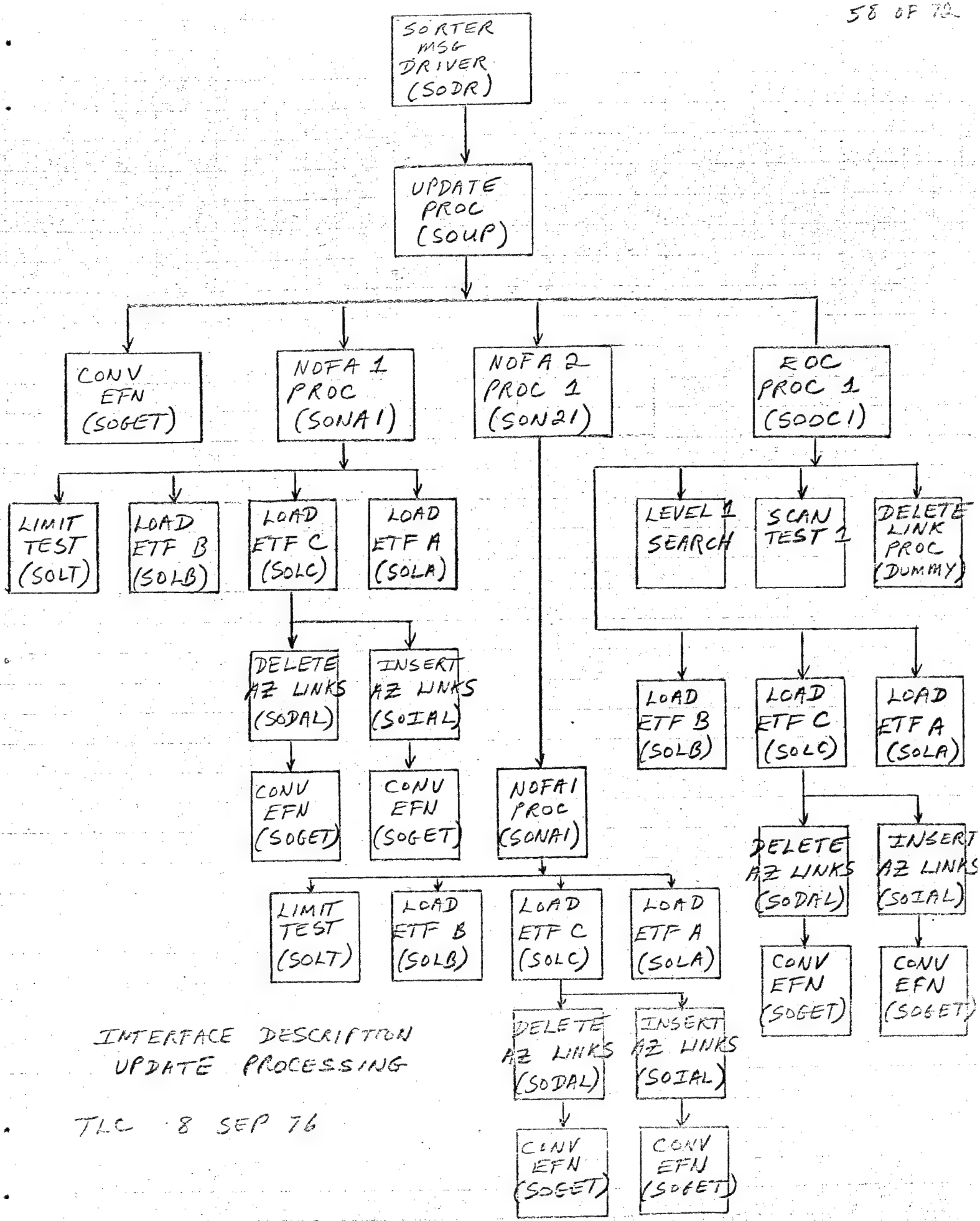
3.8.2
3.8.3
3.8.4
3.8.5
3.8.6
3.8.8
3.8.11



INTERFACE DESCRIPTION

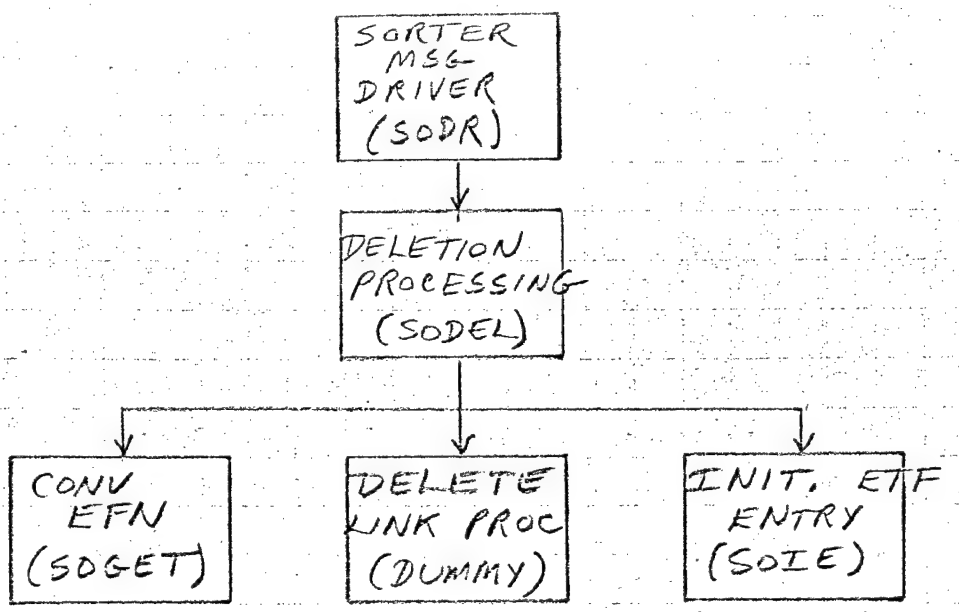
- THROTTLE ALERT PROC
- CONFIRM FILE CREATION PROC
- SYSTEM MGMT 1 PROC
- LONG PULSE PROC
- ALR-50 PROC
- MFF PROC
- SORTER INSTRUMENTATION PROC

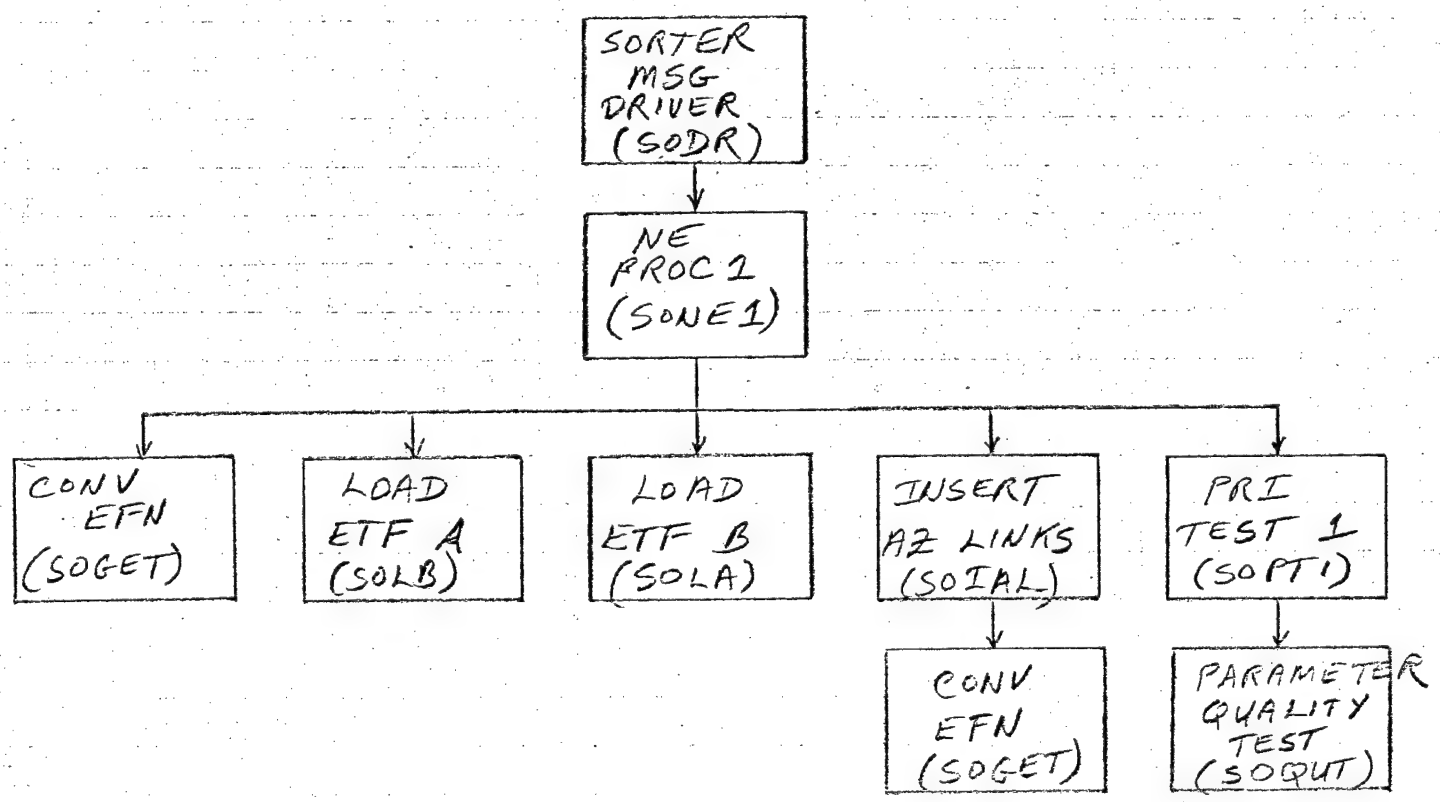
TLC 8 SEP 75



INTERFACE DESCRIPTION
UPDATE PROCESSING

TLC 8 SEP 76



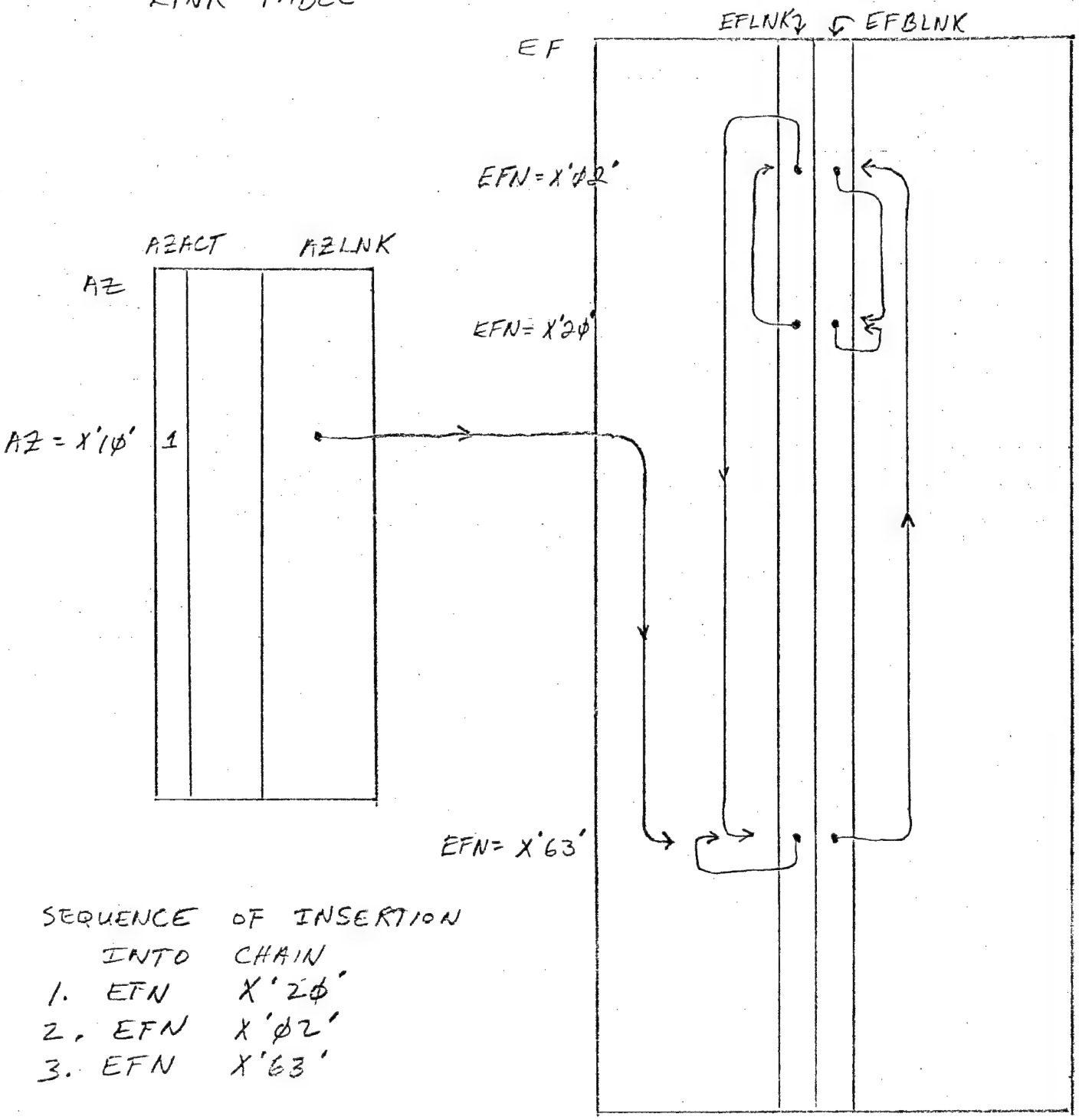


INTERFACE DESCRIPTION
— DELETION PROCESSING
— NEW EMITTER PROC 1

TLC 8 SEP 76

EMITTER TRACK FILE

AZIMUTH
LINK TABLE



SEQUENCE OF INSERTION
INTO CHAIN

1. EFN X'20'
2. EFN X'02'
3. EFN X'63'

AZIMUTH CHAIN EXAMPLE

Fig 1.

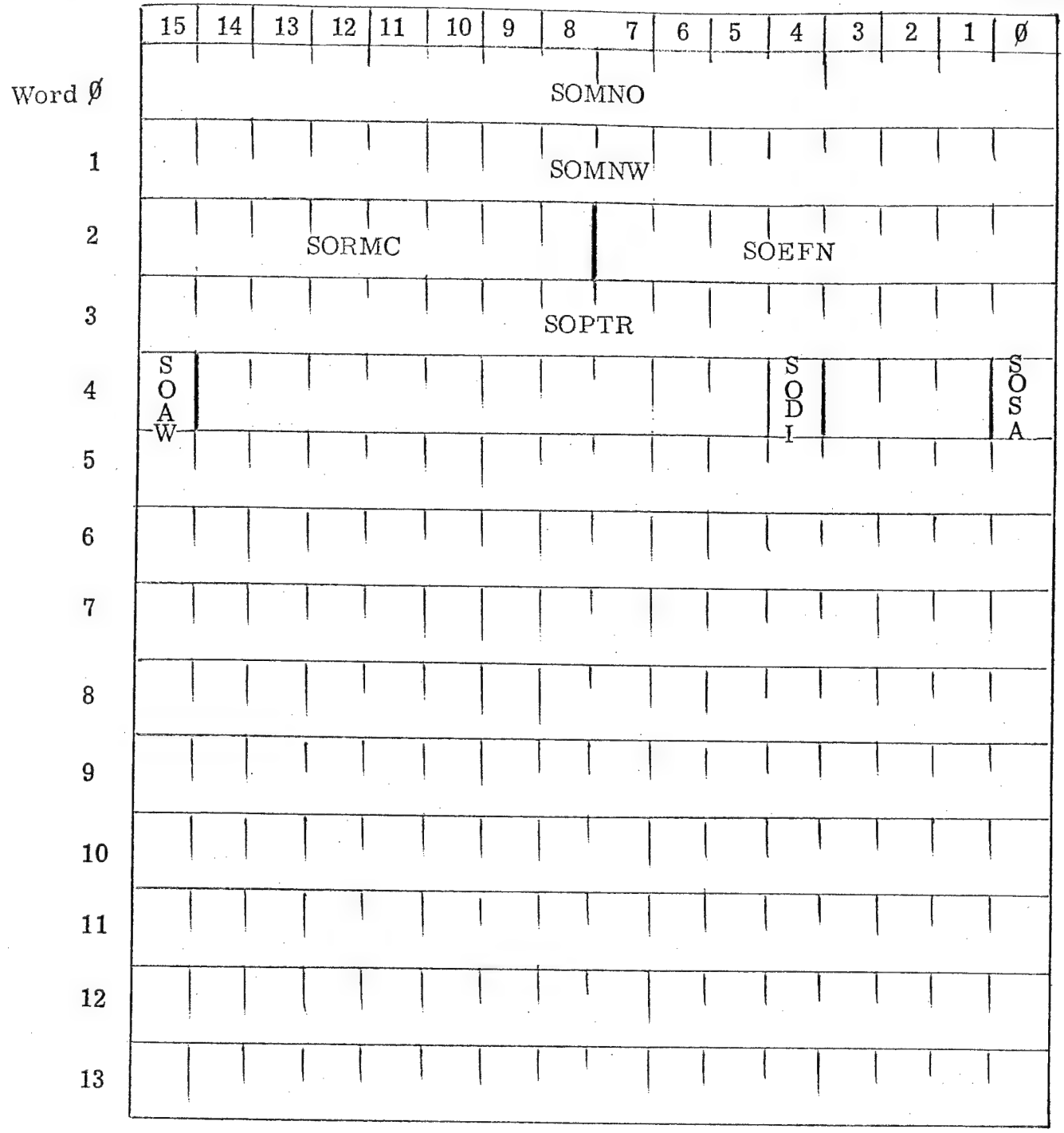


Figure 2a. Analysis Request Message

Field	Description	Units	LSB
SOMNO	Executive Message No. (=TBD)	N/A	1
SOMNW	Number of words in message (= 3)	N/A	1
SORMC	Return module code (= one of following: X'Ø3' NOFA2 Proc 2 X'Ø5' EOC Proc 2)	N/A	N/A
SOEFN	Emitter File No. ($0 \leq$ SOEFN ≤ 127)	N/A	1
SOPTR	Pointer to Candidate List (= Ø if SORMC = X'Ø3')	N/A	N/A
SOAW	Analysis Wanted Code (Ø = No Anal, 1 = Anal)	N/A	N/A
SODI	Deinterleaving Analysis Request (Ø = None, 1 = Do Deinterleaving)	N/A	N/A
SOSA	Scan Analysis Request (Ø = None, 1 = Do Scan Anal)	N/A	N/A

Figure 2b. Analysis Request Message

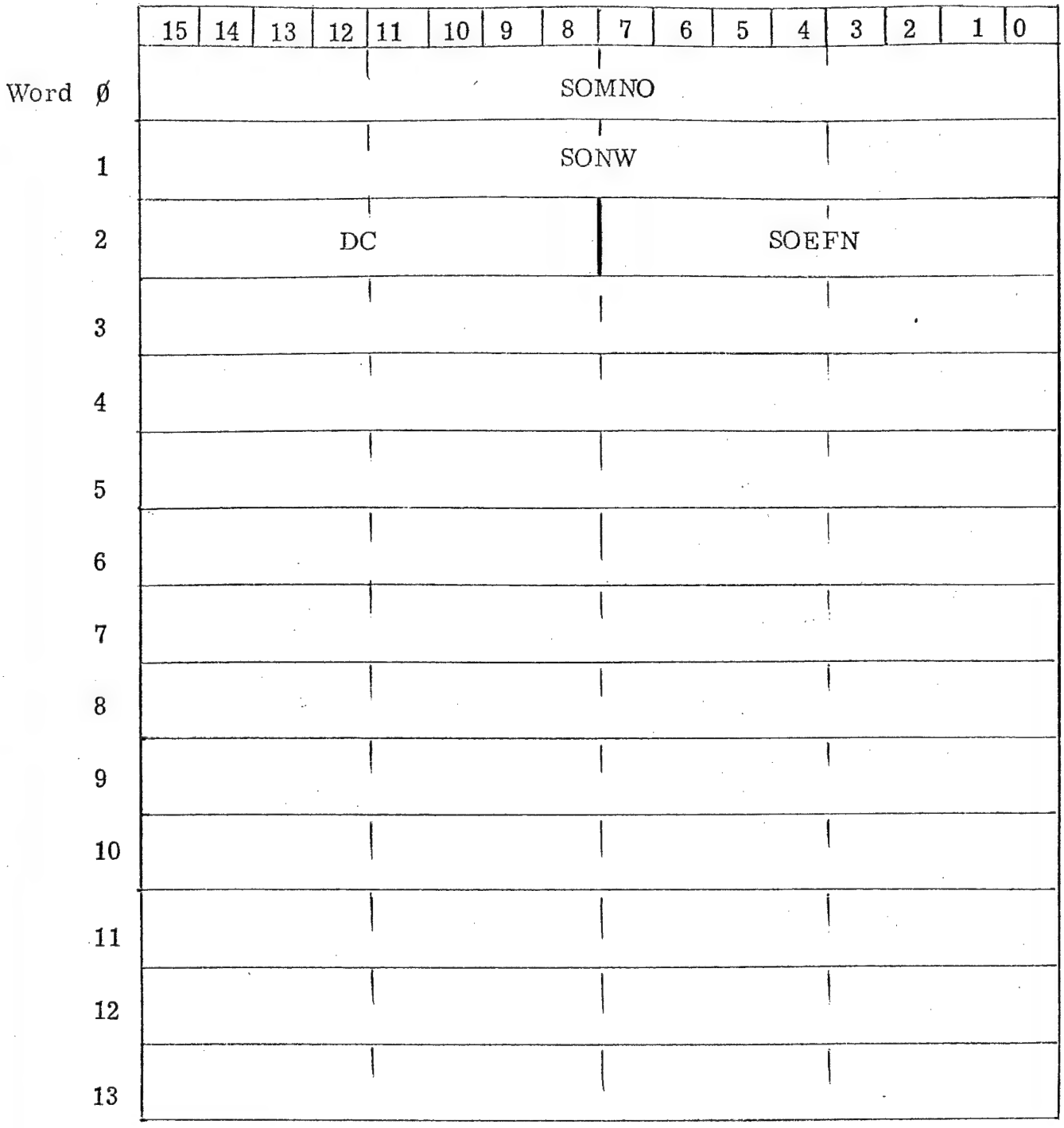


Figure 3a. Classification Message (to CP)

Field	Description	Units	LSB
SOMNO	Message Number (= TBD)	N/A	1
SOMNW	Number of words in message (= 1)	N/A	1
SOEFN	Emitter file number (Ø≤ SOEFN ≤127)	N/A	1

Figure 3b. Classification Message (to CP)

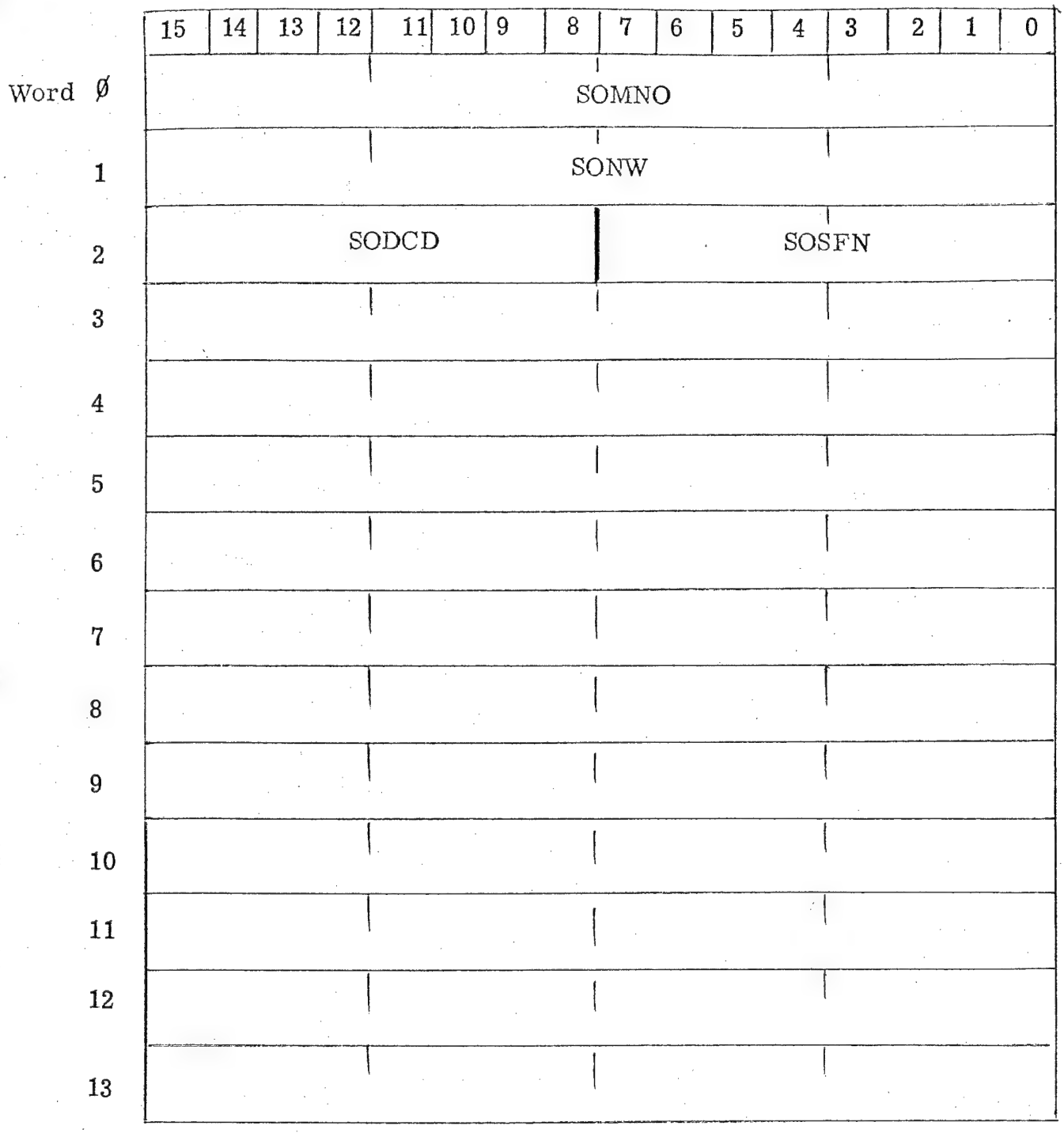


Figure 4a. Delete Track File Message (to Sorter)

Field	Description	Units	LSB
SOMNO	Message Number (= TBD)	N/A	1
SOMNW	Number of words in message (= 1)	N/A	1
SODCD	SC-to Sorter Deletion Message Op-code (= X'11')	N/A	N/A
SOSFN	Sorter file no. ($\emptyset \leq$ SOSFN $\leq$ 127)	N/A	1

Figure 4b. Delete Track File Message (to Sorter)

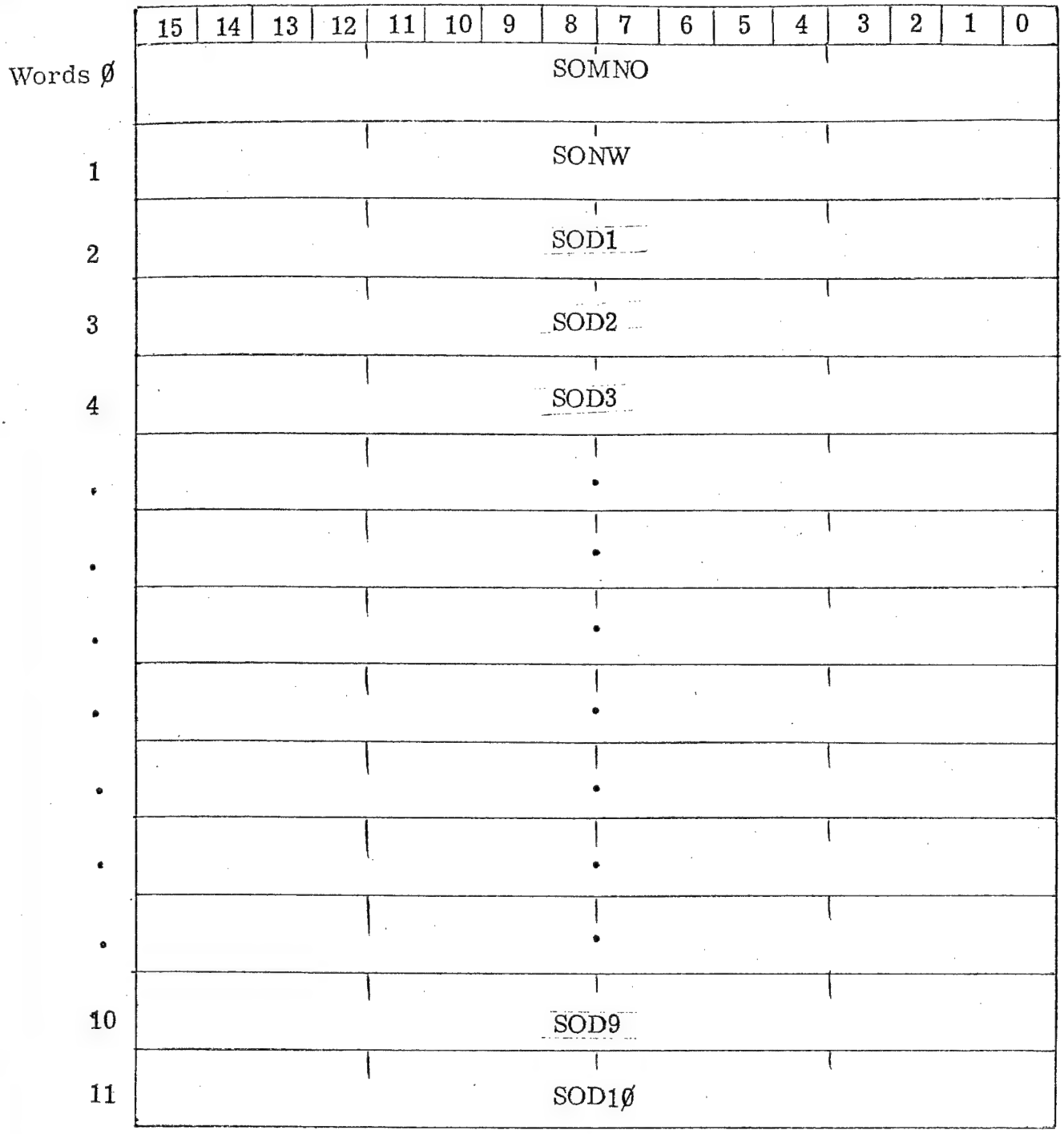


Figure 5a. Sorter Instrumentation Message Format

Field	Description	Units	LSB
SOMNO	Message Number (= TBD)	N/A	1
SONW	Number of words in message	N/A	1
SOD1	Sorter Message Word 1 (Op-Code, etc.)	N/A	N/A
SOD2	" " " 2	N/A	N/A
SOD3	" " " 3	N/A	N/A
SOD4	" " " 4	N/A	N/A
SOD5	" " " 5	N/A	N/A
SOD6	" " " 6	N/A	N/A
SOD7	" " " 7	N/A	N/A
SOD8	" " " 8	N/A	N/A
SOD9	" " " 9	N/A	N/A
SOD10	" " " 10	N/A	N/A

Figure 5b. Sorter Instrumentation Message Format

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Words Ø	SOMNO															
1	SONW															
2	SOS1															
3	SOS2															
4	SOS3															
.																
.																
.																
.																
.																
.																
.																
10	SOS9															
11	SOS1Ø															

Figure 6a. System Management 1 Message (to RMP)

Field	Description	Units	LSB
SOMNO	Message Number (= TBD)	N/A	1
SONW	Number of words in message	N/A	N/A
SOS1	Sorter Message Word 1 (Op-Code, etc.)	N/A	N/A
SOS2	" " " 2	N/A	N/A
SOS3	" " " 3	N/A	N/A
SOS4	" " " 4	N/A	N/A
SOS5	" " " 5	N/A	N/A
SOS6	" " " 6	N/A	N/A
SOS7	" " " 7	N/A	N/A
SOS8	" " " 8	N/A	N/A
SOS9	" " " 9	N/A	N/A
SOS10	" " " 10	N/A	N/A

Figure 6b. System Management 1 Message (to RMP)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Words Ø	SOMNO															
1	SONW															
22	DC											SOSFN				
3																
4																
5																
6																
7																
8																
9																
10																
11																
12																
13																

Figure 7a. Update Message (to RM)

Field	Description	Units	LSB
SOMNO	Message Number (= TBD)	N/A	1
SOMNW	Number of words in message (= 1)	N/A	1
SOSFN	Sorter File No. ($\emptyset \leq$ SOSFN $\leq$ 127)	N/A	1

Figure 7b. Update Message (to RMP)